A Content-Based Image Retrieval System (CBIR) for eCommerce Purposes Using Deep Neural Networks

Lee Reed
Stanford University
leereed@stanford.edu

Nicholas Sinthunont
Stanford University
nsinthun@stanford.edu

Abstract

This study investigates the use of deep neural networks to improve the current process of how eCommerce search is conducted. The proposed system consist of 2 components: a detection model which uses Mask R-CNN and a image matcher which uses VGG16 coupled with an approximate nearest neighbor algorithm. The results suggest that such a system is possible to implement in real world applications with significant improvement compared to the current systems used by existing businesses.

1. Introduction

In 2000, psychologists Sheena Iyengar and Mark Lepper performed an experiment which shocked marketers and retailers alike. This study is known as the Jam Study. In a upscale food supermarket, they set up a table offering 24 varieties of jam (anyone who sampled the spread would get a $1 coupon off any jam of their choice). A few days later they repeated the experiment except this time they only offered 6 varieties of jam.

The result was that the larger display with 24 jams attracted more interest but customers who saw the larger display were only 1/10th as likely to purchase as customers who saw the smaller display. The psychologists reasoned that consumers viewing the larger display suffered from what they call "choice paralysis".

Since the dawn of the digital era, eCommerce has exploded and has given rise to a larger selection of products which are on "display". Amazon, the largest eRetailer in the world, has over 500 million products in its catalogue which excludes variants of products. The strategic focus of eCommerce companies to solely focus on serving the long tail has created a similar "choice paralysis" to the Jam Study. Consumers now have to deal with 2 major issues with their online shopping experience:

Browse: Consumers who attempt to use the existing platforms to browse will be presented with pages and pages of choices, most of which may not even be related to the item they want to purchase

Search limitations: In most cultures, consumers typically begin their journey by being inspired by images they see on social media however eCommerce platforms generally still only allow for text based inputs[1]

A few solutions have been rolled out commercially to address this issue which aim to give consumers a better experience. At the forefront of these solutions is the use of AI and recommendation systems. Knowing what the consumer has browsed, it is possible to guess the exact product/item that they are interested in either using systems such as content-based filtering or collaborative filtering. However, frequently the items recommended can either be not exactly what the consumer is looking for or a completely random item

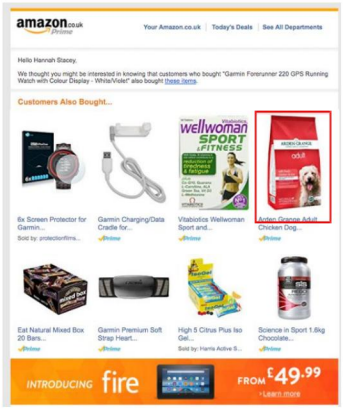

Figure [1]: Amazon recommending dog product for a user who doesn't own a dog[2]

2. Related Work

Various technology companies globally are now slowly moving to an image based search approach so to give the

user more control with regards to expressing what they are interested in. One of the most successful platforms to be released commercially is Taobao which offers users the ability search via images:

Figure [2]: Taobao image search

Other examples of image search platforms include eBay, which only released it within the last 12 months, and Pinterest.

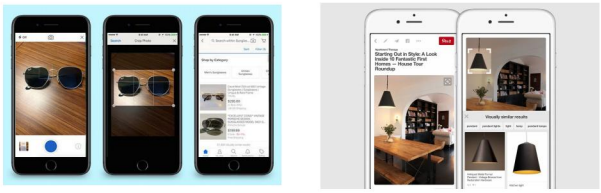

Figure [3]: eBay (left) and Pinterest(right) image search

These platforms differ in their approach by the interface they present and how much work the user has to do. For example, Taobao and eBay asks for the user to crop their images beforehand while Pinterest incorporates an object detection model which finds relevant objects and gives recommendations[3].

This paper will take a similar approach to Pinterest and adopt a system which will consist of both an object detector and an image matcher.

Object detection: The field of object detectors has grown over the last few years to become more robust and accurate. Initially only able to detect single objects, detectors are now able to not only detect multiple objects but also the exact outline/segmentation of those different objects:

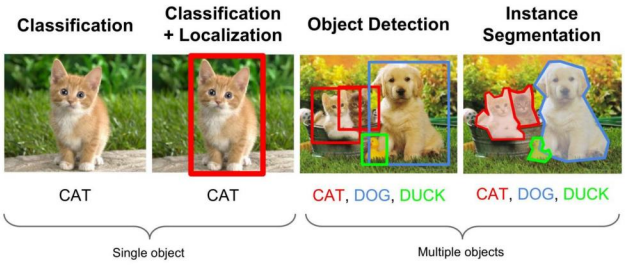

Figure [4]: Evolution of object detectors[4]

The ability to just detect or to also segment objects will depend on the type of algorithm. R-CNNs and Faster R-CNNs have proven to be robust networks which can detect objects with good accuracy while YOLO (You Only Look Once) is able to achieve real time performance. However, in order to achieve segmentation as well, Mask R-CNN is required[4]. The choice between each algorithm will depend on the desired trade-off between accuracy and speed.

Model	PASCAL VOC 2007	PASCAL VOC 2010	PASCAL VOC 2012	COCO 2015 (IoU=0.5)	COCO 2015 (IoU=0.75)	COCO 2015 (Official Metric)	COCO 2016 (IoU=0.5)	COCO 2016 (IoU=0.75)	COCO 2016 (Official Metric)	Real Time Speed
R-CNN	x	62.4%	x	x	x	x	x	x	x	No
Fast R-CNN	70.0%	68.8%	68.4%	x	x	x	x	x	x	No
Faster R-CNN	78.8%	x	75.9%	x	x	x	x	x	x	No
R-FCN	82.0%	x	x	53.2%	x	31.5%	x	x	x	No
YOLO	63.7%	x	57.9%	x	x	x	x	x	x	Yes
SSD	83.2%	x	82.2%	48.5%	30.3%	31.5%	x	x	x	No
YOLOv2	78.6%	x	x	44.0%	19.2%	21.6%	x	x	x	Yes
NASNet	x	x	x	43.1%	x	x	x	x	x	No
Mask R-CNN	x	x	x	x	x	x	62.3%	43.3%	39.8%	No

Figure [5]: Comparison of detection algorithms[5]

Image matching: Image similarity is a concept that has been researched extensively over the last 15 years. Within computer vision, the most early success was the rise of the SIFT algorithm[6], which help to discretize an image into various key points, each within a 128-dimensional space. However, deep learning has provider additional tools in the form of convolutional neural networks (CNNS) which now presents an alternative to the SIFT approach:

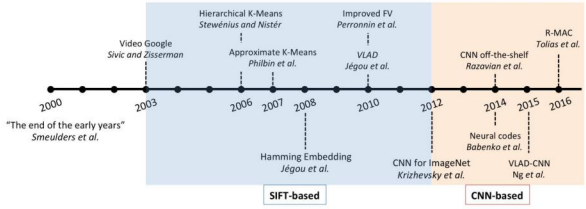

Figure [6]: history of image matching algorithms[7]

Previous research has demonstrated CNNs typically yield more accurate results (assuming sufficient training data) thus is a suitable choice for image feature extraction and comparison. The indexing within the database is then performed using an approximate nearest neighbour approach. However, the exact system may vary and incorporate different elements which could include choices between VGG vs AlexNet vs ResNet etc…

3. Method

The proposed system that this paper will explore will include 2 components, an object detector and an image matcher which will work in tandem:

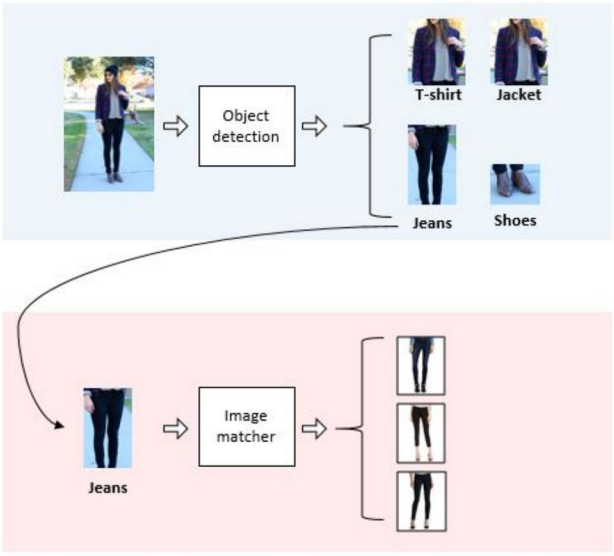

Figure [7]: Proposed system

For the object detector, an existing implementation of Mask R-CNN[8] will be utilized. Since the use case will be the ability to detect and classify "objects in the wild" (i.e. non professional photos), it will be important to feed accurate images into the image matcher. Since Mask R-CNNs are a combination of both fully convolutional networks (FCNs) and Faster RCNN, FCN alone does provide a faster method to image segmentation however at the cost of having spurious edges present. Since it is important to pass a full image of the object to the image matcher, FCNs were deemed inappropriate.

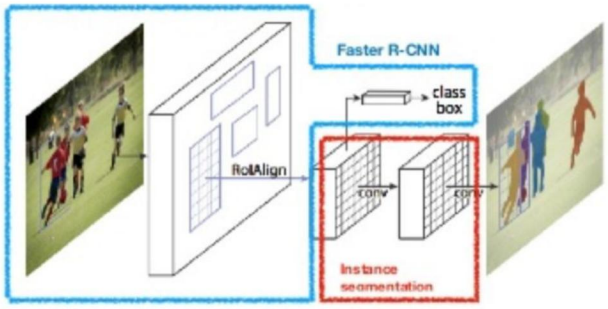

Figure [x]: Mask R-CNN network

In addition, with Mask R-CNNs, the object mask will also be provided as an output thus it will be possible to silence all the surrounding noise and feed the image matcher an exact image of the product. This feature makes it more superior in the system compared to non-segmentation detectors such as YOLO and SSD:

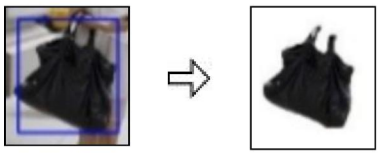

Figure [8]: Detected bag being isolated using mask

For the image matcher, a VGG16 network, which was trained on the imagenet dataset, was utilized. The output of this model would be the embeddings after the 5 stage thus would be in the shape of 7 x 7 x 512.

For each of the images in the database, the output was vectorized to create a 25,088 dimensional space representation of the input image which was stored in a KNN space. In doing so, it would be possible to push a training image through the VGG network to find the most similar image in the database

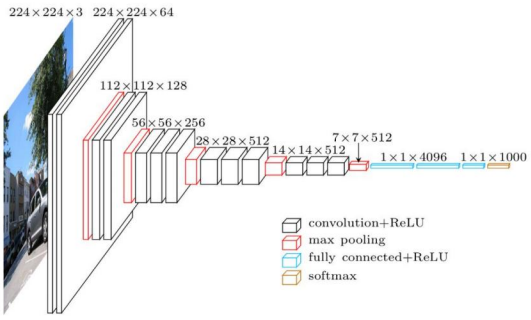

Figure [9]: VGG16 network

To determine similarity amongst images the Annoy (Approximate Nearest Neighbors Oh Yeah) was utilized to identify closest Neighbors to a given input image provided by the object detection model. Annoy provides multiple benefits from speed and performance standpoint because it is able to use static files as indices. Thus index plus associated feature vectors only needs to be saved into RAM once on initial load. Indexes are used for lookups after this point when determine approximate nearest neighbors. Every feature vector and index is represented as a separate vector in f-dimensional space.

In our project case, after loading complete set of embeddings from VGG16 output into a database we then use Euclidean distance from input image to determine top k most similar database images. These images are selected using Annoy's priority queue to search all trees until nearest k items are found. Formula to determine distance is shown below:

$$distance = sqrt(2(1 - cos(u, v)))$$

4. Data

Data was scraped from eCommerce websites such as Nordstrom and consisted of various classes including men's jeans/t-shirts/shorts and women's dresses/tees/skirts. This images were specifically chosen because they consisted of people actually wearing the article of clothing as opposed to just images of the clothing:

Figure [10]: product image without person (not desirable) and product with person (desirable)

In addition, Google's Open Images v4 was collected since it provided a dataset with over 15 million bounding boxes across 600 classes (clothing bounding boxes in the order of 10^6)[9].

5. Experiments

The experiments were conducted across both the detector and the image matcher but separately.

5.1. Detector

One of the largest constraints applying deep learning models to commercial applications is the availability of data. Therefore 3 experiments were conducted in order to increase the robustness and accuracy of the applying Mask R-CNNs.

5.1.1 Google's open image dataset

In May 2018, Google released the fourth version of their open image dataset. This dataset consist of 1.7 million images which all have bounding boxes that represent 1 of 600 classes. In addition, Google also provided an additional 41,000 validation images and 123,000 test images; all of which also have boxing box data. While not all the data provided has boxing boxes related to fashion, there was data in the order of 10^5 thus was deemed sufficient to analyze.

Figure [11]: Mask R-CNN results using Google open image dataset

The results highlighted a constraint on the Mask R-CNN algorithm which is that without mask data, it is unable to detect objects with abstract shapes. For example, the algorithm was found to be able to accurately identify "footwear" as the objects within this class were typically rectangular in nature. However, t-shirts and jeans are much more abstract in their appearance thus the algorithm performed poorly in this class.

5.1.2 Mechanical turk approach

In order to train the Mask R-CNN with more relevant data, an approach using mechanical turks was implemented. This involved manually creating masks around the training images already collected.

Figure [12]: example of creating mask of existing images

Historical attempts suggested that only a small amount of training data would be required since the Mask R-CNN had already already been trained extensively using the coco dataset. When training and testing just 1 class (e.g. jeans), the results suggested a 94% accuracy. However, when training and testing on 16 different classes (e.g. t-shirt, dress, miniskirt, shoes, jacket etc…) the accuracy was significantly reduced to 43%

Figure [13]: results of single class vs multi class

5.1.3 FCN data generation

In order to improve the results from the mechanical turk dataset, more data was needed since the multi class experiment only consisted of 500 images but with 16 classes. However, the mechanical turk process was restrictive because of time requirements therefore an alternative approach was needed.

Since mask data was scarce, it was deemed appropriate to try and generate data. This process would involve using catalogue data and pushing the images through a human parser. The human parser would generate instance segmentation maps whereby the specific item of clothing could be isolated thus a mask be generated:

Figure [14]: example of instance segmentation using a catalogue image to generate dress mask

The human parsers was built using a variant of a fully convolutional network (FCN) which allows for each portion of a person's body to be represented in isolation. However, it can be seen that sometimes the results are not ideal:

Figure [15]: example of a problem with instance segmentation

When testing the results using this new FCN generated data, the results were significantly better. After 1,500 steps, the loss using this new data set was ~33% of the loss using the Google open image set and ~50% of the mechanical turk data.

5.2. Image matcher

Overall results for similarity varied significantly depending on the class in question. In all cases find duplicate results as expected was perfect expediting data clean-up. However, it is unclear what the algorithm used to determine the based for similarity resulting in inaccurate results for certain classes vs. others. For

example handbags performed well resulting in min distance = 1.774 * 10-4 from input image.

Figure [16]: Similarity Matching - Handbag Example

Dresses on the other hand performed poorly likely due to the inability of the VGG16 to pick-up on key features related to dress and pattern. This is also likely impacted by higher amount of negative space as well such as skin tone of models and white background. This is shown in the below example:

Figure [17]: Similarity Matching - Dress Example

6. Conclusion

Overall, our project made several inroads into the use of deep neural networks to improve the current process of how eCommerce search is conducted relative to existing market methods. Specifically, on object detection front our results using FCN generation to improve upon our model is promising as a form of data augmentation of masked data combined with masked R-CNN resulted in less loss than FCNN methods. On image similarity front, further exploration is needed to achieve a higher degree of accuracy when using images of higher complexity. This will likely require VGG16 weights to be retrained further beyond imagenet set for scope of classes in question.

7. Future extensions

As a next step to improve upon our results we plan to implement and iterate upon the following:
- Full integration of Object Detection and Image Similarity models
- Rigorous testing to validate and tune model against different fashion classes e.g. men vs. women jeans
- Further experimentation with similarity search algorithm to compare Annoy performance with triplet loss approach
- Incorporating unsupervised techniques to automatically determine classes for boundary boxes
- Removal of background noise from detection output

Endnotes

1.https://www.bcg.com/publications/2017/retail-globalization-chinese-consumers-online-journey-from-discovery-to-purchase.aspx

2. https://blog.ometria.com/6-product-recommendation-fails-to-watch-out-for

3. https://labs.pinterest.com/user/themes/pinlabs/assets/paper/visual_search_at_pinterest.pdf

4. K. He, G. Gkioxari, P.. Dollar and R. Girshick. Mask R-CNN. arXiv:1703.06870v3 [cs.CV] 24 Jan 2018

5. https://medium.com/comet-app/review-of-deep-learning-algorithms-for-object-detection-c1f3d437b852

6. D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision, 2004

7. L. Zheng, Y. Yang, and Q. Tian. SIFT Meets CNN: A Decade Survey of Instance Retrieval. arXiv:1608.01807v2 [cs.CV] 23 May 2017

8. https://github.com/matterport/Mask_RCNN

9. https://storage.googleapis.com/openimages/web/index.html

10. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014.

11. M. Mirza, S. Osindero. Conditional Generative Adversarial Nets. arXiv:1411.1784v1 [cs.LG] 6 Nov 2014.

12. http://blog.fastforwardlabs.com/2016/08/12/introducing-variational-autoencoders-in-prose-and.html

13. P. Isola, J. Zhu, T. Zhou and A. Efros. Image-to-Image Translation with Conditional Adversarial Networks. arXiv:1611.07004v2 [cs.CV] 22 Nov 2017

14. O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. arXiv:1505.04597v1 [cs.CV] 18 May 2015

15. VITON: An Image-based Virtual Try-on Network

16. CS 230: Deep Learning http://cs230.stanford.edu/

17. Xiaodan Liang, Ke Gong, Xiaohui Shen, and Liang Lin, "Look into Person: Joint Body Parsing & Pose Estimation Network and A New Benchmark", T-PAMI 2018.

18. O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation

19. Z. Zhang, Q. Liu, and Y. Wang. Road Extraction by Deep Residual U-Net

20. G. E. Hinton*, N. Srivastava, A. Krizhevsky, I. Sutskever and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. arXiv:1207.0580v1 [cs.NE] 3 Jul 2012

21. H. Wu and X. Gu.Towards Dropout Training for Convolutional Neural Networks. arXiv:1512.00242 [cs.NE] Dec 2015

22. https://engineering.matterport.com/splash-of-color-instance-segmentation-with-mask-r-cnn-and-tensorflow-7c761e238b46

23.https://github.com/Engineering-Course/LIP_JPPNet