Visual Search at Pinterest

Yushi Jing[1], David Liu[1], Dmitry Kislyuk[1], Andrew Zhai[1], Jiajing Xu[1], Jeff Donahue[1,2], Sarah Tavel[1]
[1]Visual Discovery, Pinterest
[2]University of California, Berkeley
{jing, dliu, dkislyuk, andrew, jiajing, jdonahue, sarah}@pinterest.com

ABSTRACT

We demonstrate that, with the availability of distributed computation platforms such as Amazon Web Services and open-source tools, it is possible for a small engineering team to build, launch and maintain a cost-effective, large-scale visual search system. We also demonstrate, through a comprehensive set of live experiments at Pinterest, that content recommendation powered by visual search improves user engagement. By sharing our implementation details and learnings from launching a commercial visual search engine from scratch, we hope visual search becomes more widely incorporated into today's commercial applications.

Categories and Subject Descriptors

H.3.3 [**Information Systems Applications**]: Search Process; I.4.9 [**Image Processing and Computer Vision**]: Application

General Terms

information retrieval, computer vision, deep learning, distributed systems

Keywords

visual search, visual shopping, open source

1. INTRODUCTION

Visual search, or content-based image retrieval [5], is an active research area driven in part by the explosive growth of online photos and the popularity of search engines. Google Goggles, Google Similar Images and Amazon Flow are several examples of commercial visual search systems. Although significant progress has been made in building Web-scale visual search systems, there are few publications describing end-to-end architectures deployed on commercial applications. This is in part due to the complexity of real-world visual search systems, and in part due to business considerations to keep core search technology proprietary.

KDD'15, August 10-13, 2015, Sydney, NSW, Australia.

ACM 978-1-4503-3664-2/15/08 ...$15.00.
DOI: http://dx.doi.org/10.1145/2783258.2788621 .

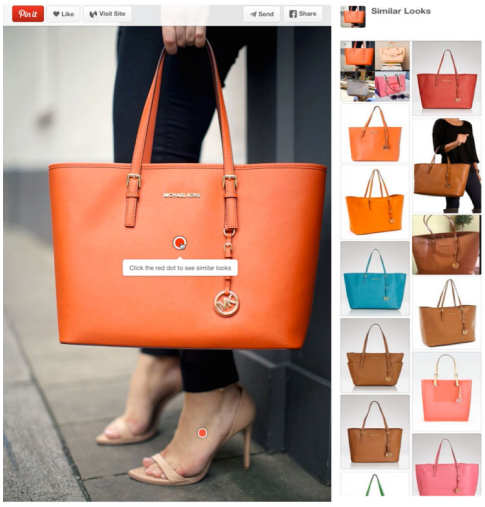

Figure 1: Similar Looks: We apply object detection to localize products such as bags and shoes. In this prototype, users click on automatically tagged objects to view similar-looking products.

We faced two main challenges in deploying a commercial visual search system at Pinterest. First, as a startup we needed to control the development cost in the form of both human and computational resources. For example, feature computation can become expensive with a large and continuously growing image collection, and with engineers constantly experimenting with new features to deploy, it was vital for our system to be both scalable and cost effective. Second, the success of a commercial application is measured by the benefit it brings to the users (e.g. improved user engagement) relative to the cost of development and maintenance. As a result, our development progress needed to be frequently validated through A/B experiments with live user traffic.

In this paper, we describe our approach to deploy a commercial visual search system with those two challenges in mind. We make two main contributions.

Our first contribution is to present our scalable and cost effective visual search implementation using widely available tools, feasible for a small engineering team to implement. Section 2.1 describes our simple and pragmatic approach to speeding up and improving the accuracy of object detection and localization that exploits the rich metadata available at Pinterest. By decoupling the difficult (and computation-

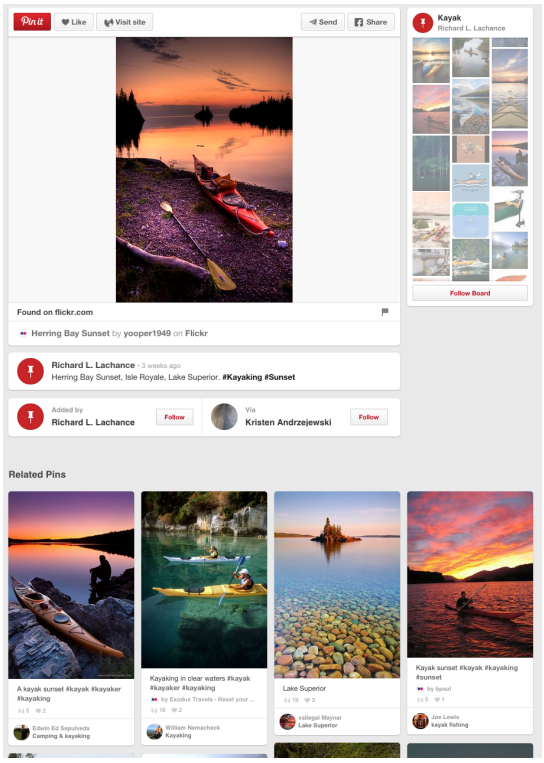

Figure 2: Related Pins is a product feature that shows recommendations based on the Pinterest curation graph.

ally expensive) task of multi-class object detection into category classification followed by per-category object detection, we only need to run (expensive) object detectors on images with a high probability of containing the object. Section 2.2 presents our distributed pipeline to incrementally add or update image features using Amazon Web Services, which avoids wasteful re-computation of unchanged image features. Section 2.3 presents our distributed indexing and search infrastructure built on top of widely available tools.

Our second contribution is to share results of deploying our visual search infrastructure in two product applications: *Related Pins* (Section 3) and *Similar Looks* (Section 4). For each application, we use application-specific data sets to evaluate the effectiveness of each visual search component (object detection, feature representations for similarity) in isolation. After deploying the end-to-end system, we use A/B tests to measure user engagement on live traffic.

Related Pins (Figure 2) is a feature that recommends Pins based on the Pin the user is currently viewing. These recommendations are primarily generated from the "curation graph" of users, boards, and Pins. However, there is a long tail of less popular Pins without recommendations. Using visual search, we generate recommendations for almost all Pins on Pinterest. Our second application, *Similar Looks* (Figure 1) is a discovery experience we tested specifically for fashion Pins. It allows users to select a visual query from regions of interest (e.g. a bag or a pair of shoes) and identifies visually similar Pins for users to explore or purchase. Instead of using the whole image, visual similarity is computed between the localized objects in the query and database images. To our knowledge, this is the first published work on object detection/localization in a commercially deployed visual search system.

Our experiments demonstrate that 1) one can achieve very low false positive rate (less than 1%) with a decent detection rate by combining the object detection/localization methods with metadata, 2) using feature representations from the VGG [22] [3] model significantly improves visual search accuracy on our Pinterest benchmark datasets, and 3) we observe significant gains in user engagement when visual search is used to power Related Pins and Similar Looks applications.

2. VISUAL SEARCH ARCHITECTURE AT PINTEREST

Pinterest is a visual bookmarking tool that helps users discover and save creative ideas. Users *pin* images to *boards*, which are curated collections around particular themes or topics. This human-curated user-board-image graph contains rich information about the images and their semantic relationships. For example, when an image is pinned to a board, it implies a "curatorial link" between the new board and all other boards the image appears in. Metadata, such as image annotations, can then be propagated through these links to describe the image, the image board and the users.

Since the image is the focus of each pin, visual features play a large role in finding interesting, inspiring and relevant content for users. In this section we describe the end-to-end implementation of a visual search system that indexes billions of images on Pinterest. We address the challenges of developing a real-world visual search system that balances cost constraints with the need for fast prototyping. We describe 1) the features that we extract from images, 2) our infrastructure for distributed and incremental feature extraction, and 3) our real-time visual search service.

2.1 Image Representation and Features

We extract a variety of features from images, including local features and "deep features" extracted from the activation of intermediate layers of deep convolutional neural networks (CNNs) [6]. We studied architectures based on AlexNet [15] and VGG [22], extracting feature representations from *fc6* and *fc8* layers. These features are binarized for representation efficiency and compared using Hamming distance. We use the open-source Caffe [12] framework to perform training and inference of our CNNs on multi-GPU machines.

The system also extracts *salient color signatures* from images. Salient colors are computed by first detecting salient regions [25, 4] of the images and then applying k-means clustering to the Lab pixel values of the salient pixels. Cluster centroids and weights are stored as the color signature of the image.

Two-step Object Detection and Localization

One feature that is particularly relevant to Pinterest is the presence of certain object classes, such as bags, shoes, watches, dresses, and sunglasses. We adopted a two-step detection approach that leverages the abundance of weak text labels on Pinterest images. Since images are pinned many times onto many boards, aggregated pin descriptions and board titles provide a great deal of information about the image.

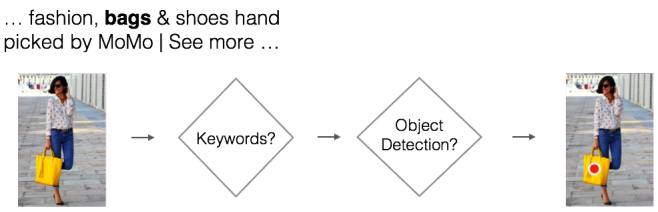

Figure 3: Instead of running all object detectors on all images, we first predict the image categories using textual metadata, and then apply object detection modules specific to the predicted category.

A text processing pipeline within Pinterest extracts relevant annotations for images from the raw text, producing short phrases associated with each image.

We use these annotations to determine which object detectors to run. In Figure 1, we first determined that the image was likely to contain bags and shoes, and then we proceeded to apply visual object detectors for those object classes. By first performing category classification, we only need to run the object detectors on images with a high prior likelihood of matching. This filtering step reduces computational cost as well as false positives.

Our initial approach for object detection was a heavily optimized implementation of cascading deformable part-based models [8]. This detector outputs a bounding box for each detected object, from which we extract visual descriptors for the object. Our recent efforts have focused on investigating the feasibility and performance of deep learning based object detectors [9, 10, 7].

Our experiment results in Section 4 show that our system achieved a very low false positive rate (less than 1%), which was vital for our application. This two-step approach also enables us to incorporate other signals into the category classification. The use of both text and visual signals for object detection and localization is widely used [2] [1] [13] for Web image retrieval and categorization.

Click Prediction

When users browse on Pinterest, they can interact with a pin by clicking to view it full screen ("close-up") and subsequently clicking through to the off-site source of the content (a click-through). For each image, we predict close-up rate (CUR) and click-through rate (CTR) based on its visual features. We trained a CNN to learn a mapping from images to the probability of a user bringing up the close-up view or clicking through to the content. Both CUR and CTR are helpful for applications like search ranking, recommendation systems and ads targeting since we often need to know which images are more likely to get attention from users.

CNNs have recently become the dominant approach to many semantic prediction tasks involving visual inputs, including classification [16, 15, 23, 3, 21, 14], detection [9, 10, 7], and segmentation [18]. Training a full CNN to learn a good representation can be time-consuming and requires a very large corpus of data. We apply transfer learning to our model by retaining the low-level visual representations from models trained for other computer vision tasks. The top layers of the network are fine-tuned for our specific task. This saves substantial training time and leverages the visual features learned from a much larger corpus than that of the target task. We use Caffe to perform this transfer learning.

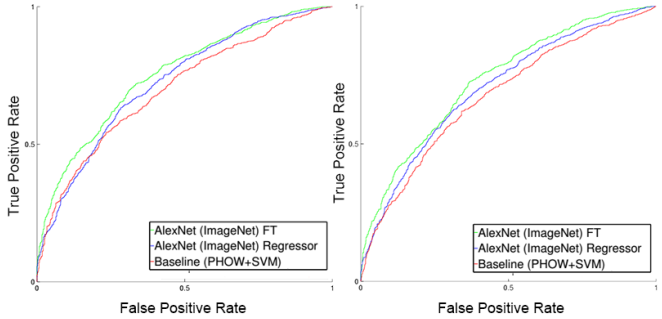

Figure 4: ROC curves for CUR prediction (left) and CTR prediction (right).

Figure 4 depicts receiver operating characteristic (ROC) curves for our CNN-based method, compared with a baseline based on a "traditional" computer vision pipeline: a SVM trained with binary labels on a pyramid histogram of words (PHOW), which performs well on object recognition datasets such as Caltech-101. Our CNN-based approach outperforms the PHOW-SVM baseline, and fine-tuning the CNN from end-to-end yields a significant performance boost as well. A similar approach was also applied to the task of detecting pornographic images uploaded to Pinterest with good results [1].

2.2 Incremental Fingerprinting Service

Most of our vision applications depend on having a complete collection of image features, stored in a format amenable to bulk processing. Keeping this data up-to-date is challenging; because our collection comprises over a billion unique images, it is critical to update the feature set incrementally and avoid unnecessary re-computation whenever possible.

We built a system called the *Incremental Fingerprinting Service*, which computes image features for all Pinterest images using a cluster of workers on Amazon EC2. It incrementally updates the collection of features under two main change scenarios: new images uploaded to Pinterest, and feature evolution (features added/modified by engineers).

Our approach is to split the image collection into *epochs* grouped by upload date, and to maintain a separate feature store for each version of each feature type (global, local, deep features). Features are stored in bulk on Amazon S3, organized by feature type, version, and date. When the data is fully up-to-date, each feature store contains all the epochs. On each run, the system detects missing epochs for each feature and enqueues jobs into a distributed queue to populate those epochs.

This storage scheme enables incremental updates as follows. Every day, a new epoch is added to our collection with that day's unique uploads, and we generate the missing features for that date. Since old images do not change, their features are not recomputed. If the algorithm or parameters

[1]By fine-tuning a network for three-class classification of *ignore*, *softcore*, and *porn* images, we are able to achieve a validation accuracy of 83.2%. When formulated as a binary classification between *ignore* and *softcore*/*porn* categories, the classifier achieved a ROC AUC score of 93.6%.

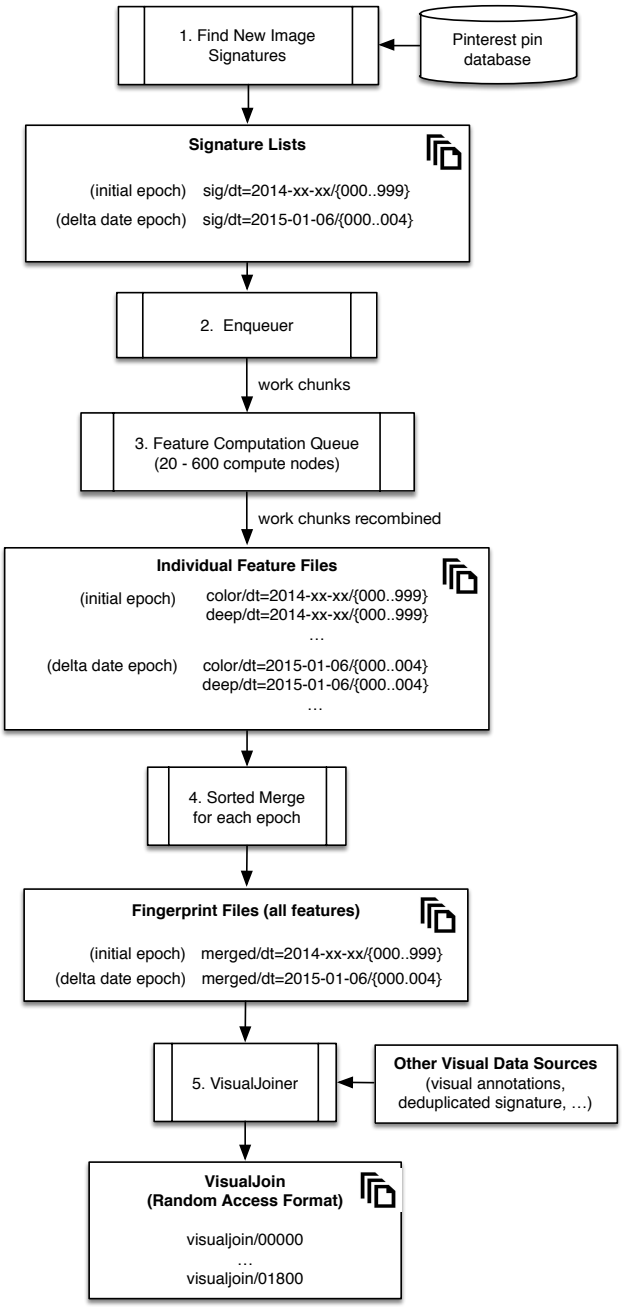

Figure 5: Examples of outputs generated by incremental fingerprint update pipeline. The initial run is shown as 2014-xx-xx which includes all the images created before that run.

for generating a feature are modified, or if a new feature is added, a new feature store is started and all of the epochs are computed for that feature. Unchanged features are not affected.

We copy these features into various forms for more convenient access by other jobs: features are merged to form a *fingerprint* containing all available features of an image, and fingerprints are copied into sharded, sorted files for random access by image signature (MD5 hash). These joined fingerprint files are regularly re-materialized, but the expensive feature computation needs only be done once per image.

A flow chart of the incremental fingerprint update process is shown in Figure 5. It consists of five main jobs: job (1) compiles a list of newly uploaded image signatures and groups them by date into epochs. We randomly divide each epoch into sorted shards of approximately 200,000 images to limit the size of the final fingerprint files. Job (2) identifies missing epochs in each feature store and enqueues jobs into PinLater (a distributed queue service similar to Amazon SQS). The jobs subdivide the shards into "work chunks", tuned such that each chunk takes approximate 30 minutes to compute. Job (3) runs on an automatically-launched cluster of EC2 instances, scaled depending on the size of the update. Spot instances can be used; if an instance is terminated, its job is rescheduled on another worker. The output of each work chunk is saved onto S3, and eventually recombined into feature files corresponding to the original shards.

Job (4) merges the individual feature shards into a unified fingerprint containing all of the available features for each image. Job (5) merges fingerprints from all epochs (along with other metadata) into a sorted, sharded HFile format allowing for random access *(VisualJoins)*.

The initial computation of all available features on all images, takes a little over a day using a cluster of several hundred 32-core machines, and produces roughly 5 TB of feature data. The steady-state requirement to process new images incrementally is only about 5 machines.

2.3 Search Infrastructure

At Pinterest, there are several use cases for a distributed visual search system. One use case is to explore similar looking products (Pinterest Similar Looks), and others include near-duplicate detection and content recommendation. In all these applications, visually similar results are computed from distributed indices built on top of the VisualJoins generated in the previous section. Since each use case has a different set of performance and cost requirements, our infrastructure is designed to be flexible and re-configurable. A flow chart of the search infrastructure is shown in Figure 6.

As the first step we create distributed image indices from VisualJoins using Hadoop. Each machine contains indexes (and features) associated with a randomly sharded subset of the entire image collection. Two types of indexes are used: the first is a disk-based (and partially memory cached) *token index* associating each vector-quantized feature (i.e. visual vocabulary token) with a posting list of image document IDs. This is analogous to a text-based image retrieval system, except text is replaced by visual tokens. The second index is an in-memory store of visual features and metadata such as image annotations and "topic vectors" computed from the user-board-image graph. The first part is used for fast (but imprecise) lookup, and the second part is used for more accurate (but slower) ranking refinement.

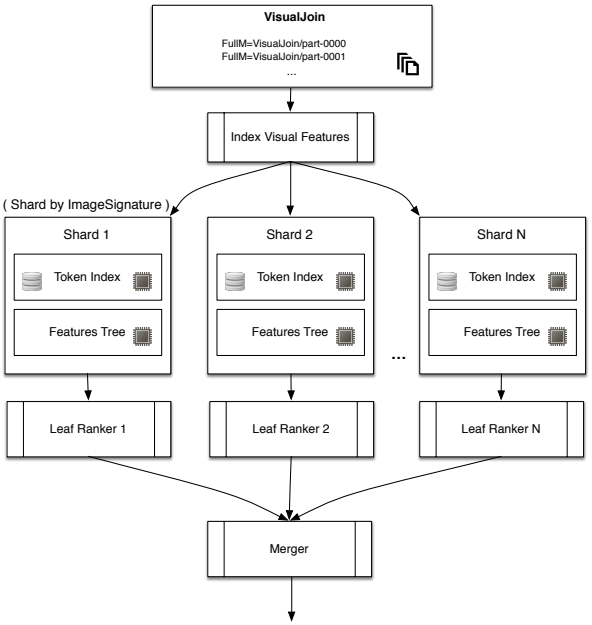

Figure 6: A flow chart of the distributed visual search pipeline.

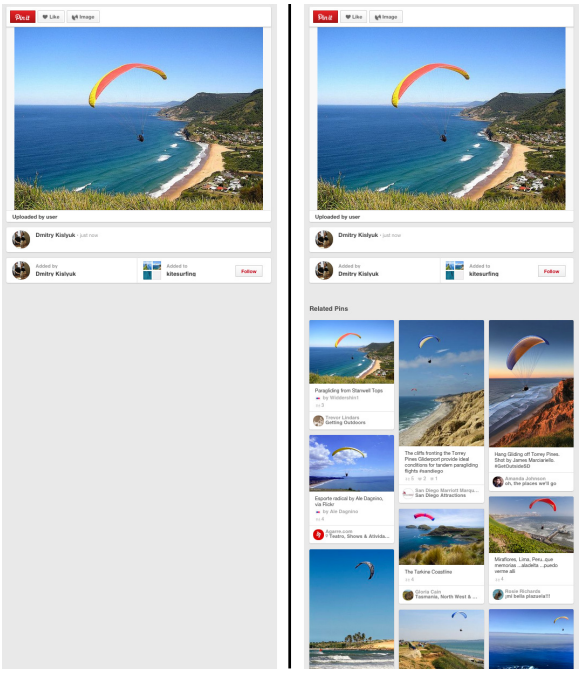

Figure 7: Before and after incorporating Visual Related Pins

Each machine runs a leaf ranker, which computes the k nearest neighbors from the indices using visual features, and then re-ranks the top candidates using additional metadata. In some cases, the leaf ranker skips the token index and directly retrieves the k nearest neighbor images from the feature tree index using approximate KNN methods such as [19]. A root ranker hosted on another machine will retrieve the top results from each of the leaf rankers, merge the results, and return them to the users. To handle new fingerprints generated with our real-time feature extractor, we have an online version of the visual search pipeline where a very similar process occurs. With the online version however, the given fingerprint is queried on pre-generated indices.

3. APPLICATION 1: RELATED PINS

One of the first applications of Pinterest's visual search pipeline was within a recommendations product called Related Pins, which recommends other images a user may be interested in when viewing a Pin. Traditionally, we have used a combination of user-curated image-to-board relationships and content-based signals to generate these recommendations. However, this system is unable to provide recommendations for less popular Pins (which do not have many connections) and newly created Pins (which may not have been indexed yet). As a result, 6% of images at Pinterest have very few or no recommendations. For these images, we used our visual search pipeline to generate *Visual* Related Pins in real time, as shown in Figure 7.

The first step of the Visual Related Pins product is to use the local token index built from all existing Pinterest images to detect if we have near duplicates to the query image. Specifically, given a query image, the system returns a set of images that are variations of the same image but altered through transformation such as resizing, cropping, rotation, translation, adding, deleting and altering minor parts of the visual content. Since the resulting images look visually identical to the query image, their recommendations are most likely relevant to the query image. In most cases, however, we found that there are either no near duplicates detected or the near duplicates do not have enough recommendations. Thus, we focused most of our attention on retrieving visual search results generated from an index based on CNN features.

Static Evaluation of Search Relevance

Our initial Visual Related Pins experiment utilized features from the original and fine-tuned versions of the AlexNet model in its search infrastructure. However, recent successes with deeper CNN architectures for classification led us to investigate the performance of feature sets from a variety of CNN models.

To conduct evaluation for visual search, we used the image annotations associated with the images as proxy for relevancy. This approach is commonly used for offline evaluation of visual search systems [20] in addition to human evaluation. In this work, we used top text queries associated with each image as labels. We retrieved 3,000 images per query for 1,000 queries using Pinterest Search, which yielded a dataset with about 1.6 million unique images. We labeled each image with the query that produced it. A visual search result is assumed to be relevant to a query image if the two images share a label.

With this dataset, we computed *precision@k* for recommendations based on several features: the *fc6* layer activations from the generic AlexNet model (pre-trained for ILSVRC) [15], the *fc6* activations of an AlexNet model fine-tuned to recognize over 3,000 Pinterest product categories, the *loss3/classifier* activations of a generic GoogLeNet [23],

Table 1: Relevance of visual search.

Model	p@5	p@10	latency
Generic AlexNet	0.051	0.040	193ms
Pinterest AlexNet	0.234	0.210	234ms
Generic GoogLeNet	0.223	0.202	1207ms
Generic VGG-16	0.302	0.269	642ms

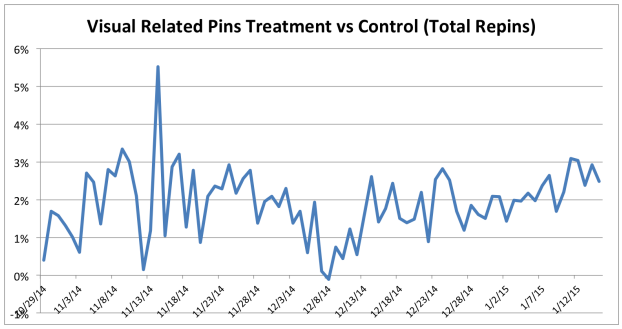

Figure 8: Visual Related Pins increases total Related Pins repins on Pinterest by 2%.

and the *fc6* activations of a generic VGG 16-layer model [3]. Table 1 shows p@5 and p@10 performance of these models, along with the average CPU-based latency of our visual search service, which includes feature extraction for the query image as well as retrieval. Using GPU-based inference dramatically reduces these latencies. We observed a substantial gain in precision against our evaluation dataset when using the FC6 features of the VGG 16-layer model, with an acceptable latency for our applications.

Live Experiments

We set up a system to detect new Pins with few recommendations, query our visual search system, and store their results in HBase to serve during Pin close-up.

For this application, we show visual search results when a majority share a common category (*category conformity thresholding*). We chose to trade off coverage for greater precision in this manner to avoid using visual search when we have relatively low confidence in its results.

We initially launched the experiment to 10% of eligible live traffic; users were eligible when they viewed a Pin close-up that did not have enough recommendations. Eligible users were triggered into either a treatment group (in which we replaced the Related Pins section with visual search results), or a control group (in which we did not alter the experience). We measured the change in total repin activity[2] within the Related Pins section.

By displaying visually similar pins for just the 6% of queries that would otherwise have empty recommendations, we observed a **2% increase** in *total* re-pin actions on Related Pins (Figure 8). Furthermore, we performed another experiment in which we reranked all recommendations using deep CNN feature similarity, achieving a **10% increase** in re-pin and click-through engagement.

[2]Repinning is the action of adding an existing Pinterest Pin to a board. Repins are one of our standard top-line metrics for measuring engagement.

4. APPLICATION 2: SIMILAR LOOKS

One of the most popular categories on Pinterest is women's fashion. However, a large percentage of pins in this category do not direct users to a shopping experience, and therefore are not actionable. There are two challenges towards making these Pins actionable: 1) Many pins feature editorial shots such as "street style" outfits, which often link to a website with little additional information on the items featured in the image; 2) Pin images often contain multiple objects (e.g. a woman walking down the street, with a leopard-print bag, black boots, sunglasses, torn jeans, etc.) A user looking at the Pin might be interested in learning more about the bag, while another user might want to buy the sunglasses.

User research revealed this to be a common frustration, and our data indicated that users are much less likely to click through to the external Website on women's fashion Pins, relative to other categories.

To address this problem, we built a product called "Similar Looks", which localizes and classifies fashion objects (Figure 9). We use object recognition to detect products such as bags, shoes, pants, and watches in Pin images. From these objects, we extract visual and semantic features to generate product recommendations ("Similar Looks"). A user would discover the recommendations via a red dot displayed on the object in the Pin (see Figure 1). Clicking on the red dot loads a feed of Pins featuring visually similar objects (e.g. other visually similar blue dresses).

Related Work

Applying visual search to "soft goods" has been explored both within academia and industry. Like.com, Google Shopping, and Zappos (owned by Amazon) are a few well-known applications of computer vision to fashion recommendations. Baidu and Alibaba also recently launched visual search systems targeting similar problems. There is also a growing amount of research on vision-based fashion recommendations [24, 17, 11]. Our approach demonstrates the feasibility of an object-based visual search system for tens of millions of Pinterest users and exposes an interactive search experience around these detected objects.

Static Evaluation of Object Localization

The first step of evaluating our Similar Looks product was to investigate our object localization and detection capabilities. We chose to focus on fashion objects because of the aforementioned business need and because "soft goods" tend to have distinctive visual shapes (e.g. shorts, bags, glasses).

We collected our evaluation dataset by randomly sampling a set of images from Pinterest's women's fashion category, and manually labeling 2,399 fashion objects in 9 categories (shoes, dress, glasses, bag, watch, pants, shorts, bikini, earnings) on the images by drawing a rectangular crop over the objects. We observed that shoes, bags, dresses and pants were the four largest categories in our evaluation dataset. Shown in Table 2 is the distribution of fashion objects as well as the detection accuracies from the text-based filter, image-based detection, and the combined approach (where text filters are applied prior to object detection).

As previously described, the text-based approach applies manually crafted rules, such as regular expressions, to the Pinterest metadata associated with images (which we treat as weak labels). For example, an image annotated with "spring fashion, *tote* with flowers" will be classified as "bag,"

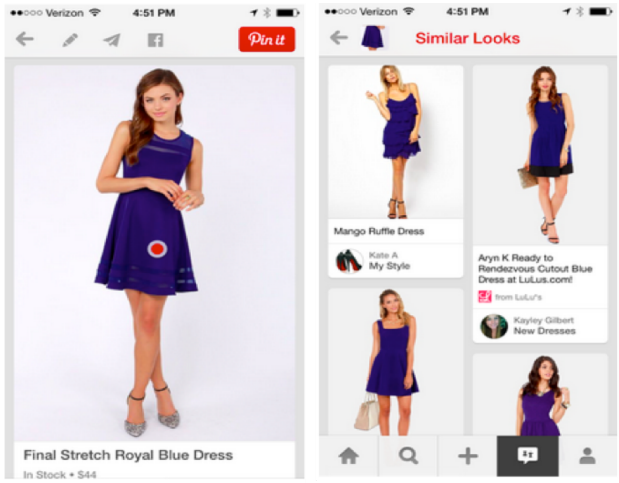

Figure 9: Once a user clicks on the red dot, the system shows products that have a similar appearance to the query object.

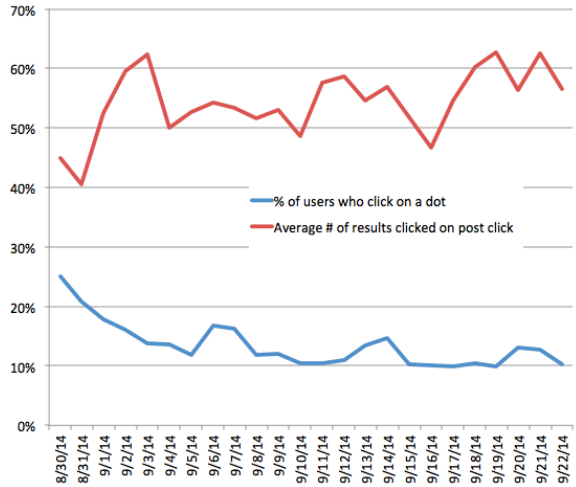

Figure 10: Engagement rates for Similar Looks experiment

Table 2: Object detection/classification accuracy (%)

		Text		Img		Both	
Objects	#	TP	FP	TP	FP	TP	FP
shoe	873	79.8	6.0	41.8	3.1	34.4	1.0
dress	383	75.5	6.2	58.8	12.3	47.0	2.0
glasses	238	75.2	18.8	63.0	0.4	50.0	0.2
bag	468	66.2	5.3	59.8	2.9	43.6	0.5
watch	36	55.6	6.0	66.7	0.5	41.7	0.0
pants	253	75.9	2.0	60.9	2.2	48.2	0.1
shorts	89	73.0	10.1	44.9	1.2	31.5	0.2
bikini	32	71.9	1.0	31.3	0.2	28.1	0.0
earrings	27	81.5	4.7	18.5	0.0	18.5	0.0
Average		72.7	6.7	49.5	2.5	38.1	0.5

and is considered as a positive sample if the image contains a "bag" object box label. For image-based evaluation, we compute the intersection between the predicted object bounding box and the labeled object bounding box of the same type, and count an intersection over union ratio of 0.3 or greater as a positive match.

Table 2 demonstrates that neither text annotation filters nor object localization alone were sufficient for our detection task due to their relatively high false positive rates at 6.7% and 2.5% respectively. Not surprisingly, combining two approaches significantly decreased our false positive rate to less than 1%.

Specifically, we saw that for classes like *glasses*, text annotations were insufficient and image-based classification excelled (likely due to the distinctive shape of glasses). For other classes, such as *dress*, this situation was reversed (the false positive rate for our dress detector was high, 12.3%, due to occlusion and high variance in style for that class, and adding a text filter dramatically improved results). Aside from reducing the number of images we needed to process with our object classifiers, for several object classes (*shoe*, *bag*, *pants*), we observed that text filtering was crucial to achieve an acceptable false positive rate (under 1%).

Live Experiments

Our system identified over 80 million "clickable" objects from a subset of Pinterest images. A clickable red dot is placed upon the detected object. Once the user clicks on the dot, our visual search system retrieves a collection of Pins with other instances of the object class, ranked by visually similarity to the query object. We launched the system to a small percentage of Pinterest live traffic and collected user engagement metrics for one month. Specifically, we looked at the clickthrough rate (CTR) of the dot, the CTR on our visual search results, and we compared engagement on Similar Look results with the engagement on existing Related Pins recommendations.

As shown in Figure 10, on average, 12% of users who viewed a pin with a dot clicked on a dot in a given day. Those users went on to click an average of 0.55 Similar Looks results. Although this data was encouraging, when we compared engagement with all related content on the pin close-up (summing both engagement with Related Pins and Similar Look results for the treatment group, and just Related Pins engagement for the control), Similar Looks actually hurt overall engagement on the pin close-up by 4%. After the novelty effort wore off, we saw a gradual decrease in CTR on the red dots, which stabilized at around 10%.

To test the relevance of our Similar Looks results independently of the newly introduced UI (the clickable object dots), we ran an experiment in which we blended Similar Looks results directly into the existing Related Pins. This gave us a way to directly measure whether users found our visually similar recommendations more relevant than our existing non-visual recommendations. On pins where we detected an object, this experiment increased overall engagement (repins and close-ups) in Related Pins by 5%. Although we set an initial static blending ratio for this experiment (one visually similar result for every three non-visual results), we later adjusted this ratio dynamically using user click data.

5. CONCLUSION AND FUTURE WORK

We demonstrate that, with the availability of distributed computational platforms such as Amazon Web Services and

Figure 12: Samples of object detection and localization results for bags. [Green: ground truth, blue: detected objects.]

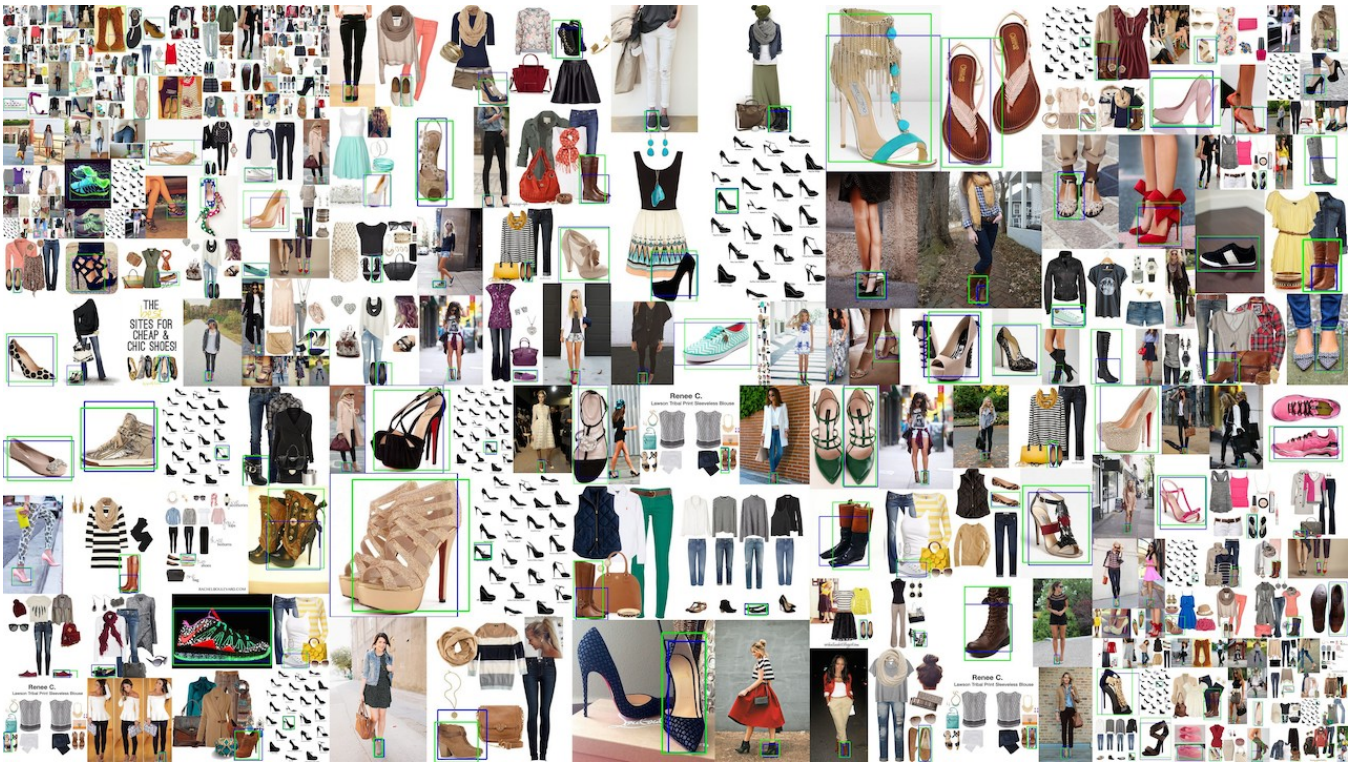

Figure 13: Samples of object detection and localization results for shoes.

Figure 14: Samples of object detection and localization results for dresses

open-source tools, it is possible for a handful of engineers or an academic lab to build a large-scale visual search system using a combination of non-proprietary tools. This paper presented our end-to-end visual search pipeline, including incremental feature updating and two-step object detection and localization method that improves search accuracy and reduces development and deployment costs. Our live product experiments demonstrate that visual search features can increase user engagement.

We plan to further improve our system in the following areas. First, we are interested in investigating the performance and efficiency of CNN based object detection methods in the context of live visual search systems. Second, we are interested in leveraging Pinterest "curation graph" to enhance visual search relevance. Lastly, we want to experiment with alternative interactive interfaces for visual search.

6. REFERENCES

[1] S. Bengio, J. Dean, D. Erhan, E. Ie, Q. V. Le, A. Rabinovich, J. Shlens, and Y. Singer. Using web co-occurrence statistics for improving image categorization. *CoRR*, abs/1312.5697, 2013.

[2] T. L. Berg, A. C. Berg, J. Edwards, M. Maire, R. White, Y.-W. Teh, E. Learned-Miller, and D. A. Forsyth. Names and faces in the news. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 848–854, 2004.

[3] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference*, 2014.

[4] M. Cheng, N. Mitra, X. Huang, P. H. S. Torr, and S. Hu. Global contrast based salient region detection. *Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 2014.

[5] R. Datta, D. Joshi, J. Li, and J. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Survey*, 40(2):5:1–5:60, May 2008.

[6] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *CoRR*, abs/1310.1531, 2013.

[7] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. Scalable object detection using deep neural networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 2155–2162, 2014.

[8] P. F. Felzenszwalb, R. B. Girshick, and D. A. McAllester. Cascade object detection with deformable part models. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2241–2248, 2010.

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *arXiv preprint arXiv:1311.2524*, 2013.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, pages 346–361. Springer, 2014.

[11] V. Jagadeesh, R. Piramuthu, A. Bhardwaj, W. Di, and N. Sundaresan. Large scale visual recommendations from street fashion images. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 14, pages 1925–1934, 2014.

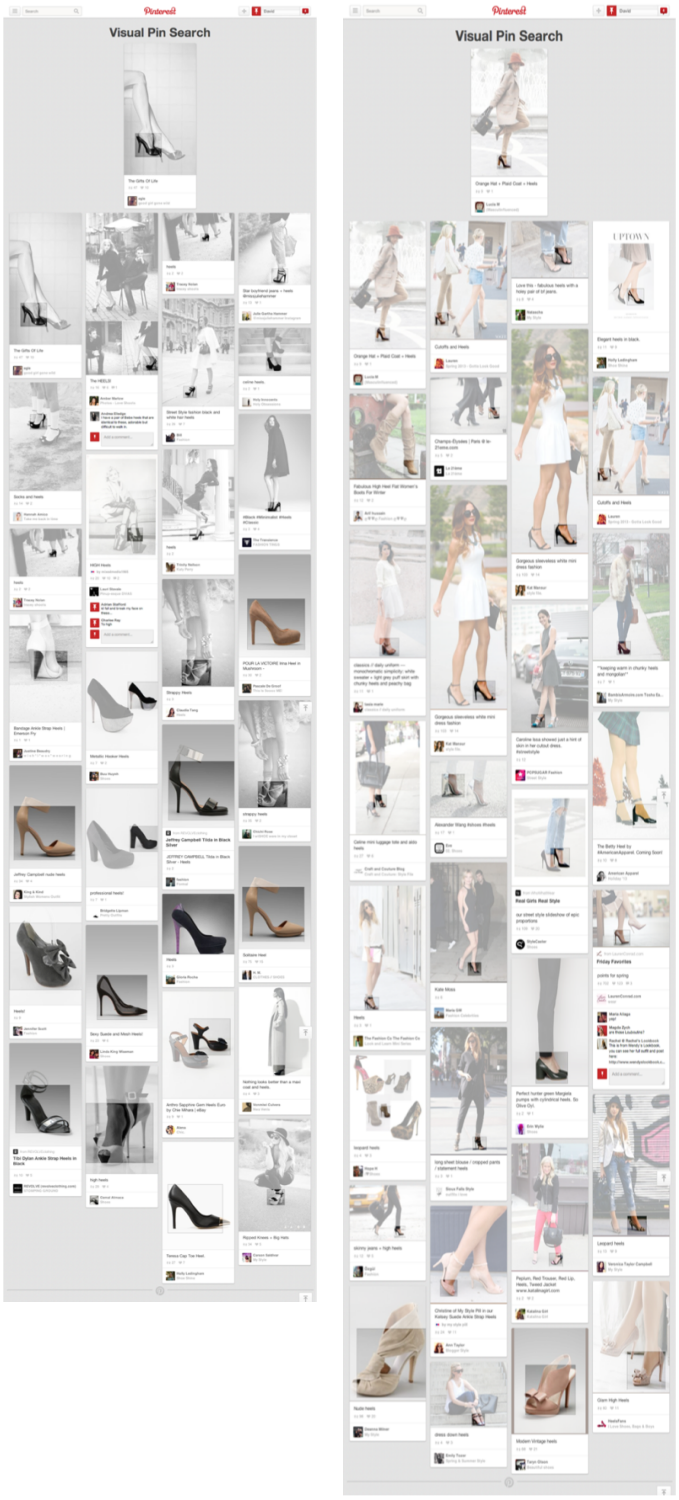

Figure 11: Examples of object search results for shoes. Boundaries of detected objects are automatically highlighted. The top image is the query image.

[12] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[13] Y. Jing and S. Baluja. Visualrank: Applying pagerank to large-scale image search. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 30(11):1877–1890, 2008.

[14] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1725–1732, 2014.

[15] A. Krizhevsky, S. Ilya, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1097–1105. 2012.

[16] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Comput.*, 1(4):541–551, Dec. 1989.

[17] S. Liu, Z. Song, M. Wang, C. Xu, H. Lu, and S. Yan. Street-to-shop: Cross-scenario clothing retrieval via parts alignment and auxiliary set. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[18] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. *arXiv preprint arXiv:1411.4038*, 2014.

[19] M. Muja and D. G. Lowe. Fast matching of binary features. In *Proceedings of the Conference on Computer and Robot Vision (CRV)*, 12, pages 404–410, Washington, DC, USA, 2012. IEEE Computer Society.

[20] H. Müller, W. Müller, D. M. Squire, S. Marchand-Maillet, and T. Pun. Performance evaluation in content-based image retrieval: Overview and proposals. *Pattern Recognition Letter*, 22(5):593–601, 2001.

[21] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. ImageNet large scale visual recognition challenge. *arXiv preprint arXiv:1409.0575*, 2014.

[22] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[23] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014.

[24] K. Yamaguchi, M. H. Kiapour, L. Ortiz, and T. Berg. Retrieving similar styles to parse clothing. *Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 2014.

[25] Q. Yan, L. Xu, J. Shi, and J. Jia. Hierarchical saliency detection. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 13, pages 1155–1162, Washington, DC, USA, 2013.