Scalable Detection of Offensive and Non-compliant Content / Logo in Product Images

Shreyansh Gandhi, Samrat Kokkula, Abon Chaudhuri, Alessandro Magnani, Theban Stanley,
Behzad Ahmadi, Venkatesh Kandaswamy, Omer Ovenc
Walmart Labs
sgandhi@walmartlabs.com

Shie Mannor
Technion
shie@ee.technion.ac.il

Abstract

In e-commerce, product content, especially product images have a significant influence on a customer's journey from product discovery to evaluation and finally, purchase decision. Since many e-commerce retailers sell items from other third-party marketplace sellers besides their own, the content published by both internal and external content creators needs to be monitored and enriched, wherever possible. Despite guidelines and warnings, product listings that contain offensive and non-compliant images continue to enter catalogs. Offensive and non-compliant content can include a wide range of objects, logos, and banners conveying violent, sexually explicit, racist, or promotional messages. Such images can severely damage the customer experience, lead to legal issues, and erode the company brand. In this paper, we present a computer vision driven offensive and non-compliant image detection system for extremely large image datasets. This paper delves into the unique challenges of applying deep learning to real-world product image data from retail world. We demonstrate how we resolve a number of technical challenges such as lack of training data, severe class imbalance, fine-grained class definitions etc. using a number of practical yet unique technical strategies. Our system combines state-of-the-art image classification and object detection techniques with budgeted crowdsourcing to develop a solution customized for a massive, diverse, and constantly evolving product catalog.

1. Introduction

By nature, humans are visual learners. A single snapshot of a product provides more information about the product than a wall of text. According to a research from "Nielson Norman"[20], only 16% of the readers actually read word-for-word and 79% only gloss over the highlights. In e-commerce, good quality images help customers understand the product better, motivate them to read about it, and build customers' trust in the product quality. This eventually increases the chance of actual purchase by the customer.

Despite the well-known importance of images, e-commerce retailers, especially the ones who allow marketplace items from 3^{rd} party sellers, struggle to control image quality. Both external and internal content creators are expected to meet the retailer's Trust & Safety guidelines. However, these guidelines constantly change and expand, which makes it incredibly difficult for e-commerce retailers to ensure that external content providers are complying with guidelines. This is why e-commerce retailers are interested to automate the process of content validation and filtering using computer vision and related technology.

Figure 1. Examples of offensive/non-compliant content : i) nudity ii) sexually explicit iii) assault rifle iv) toy resembling assault rifle)

Trust & Safety guidelines usually encompass following broad categories:

1. *Offensive Images*: Figure 1 shows various types of offensive images. The examples include images that have nudity, sexually explicit content, abusive text, objects used to promote violence, and racially inappropriate content.
2. *Non-compliant Images*: Most e-commerce retailers have published compliance guidelines on what can be

Figure 2. Which product would you choose? Promotional logos such as "best seller" is considered non-compliant.

Figure 3. Examples of marketing badges (includes award badges, the seal of excellence, best-price guarantee, lowest price, made in USA, manufactured/assembled in USA, etc.)

sold on their platform. Figure 1 [iii] and [iv] shows images of products that are non-compliant such as assault rifles and a toy that resembles assault-style rifle.

3. *Logos and Badges* : A wide range of logos and banners are considered non-compliant too. In Figure 2, the image located second from the left uses a self proclaimed marketing logo to lure the customer to click on it. This is a common malpractice and such logos are considered non-compliant. Other non-compliant logo types include competitors' logos, inaccurate manufacturing country logos (e.g., Made in USA logo), and many others (as seen in Figure 3).

Traditionally, the retailers try to address this problem either by displaying a disclaimer on the website that the displayed content is not owned by the retailer, or by allowing the customers to report non-compliant content so that they can be filtered by a human workforce. Unfortunately, these options do not protect the customer from having an unpleasant experience from seeing such images. Also, the disclaimer often goes unnoticed and the retailers brand value is tarnished. Most importantly, these solutions do not scale.

In this paper, we present a computer vision based system that automates the image detection and filtering process for an extremely large catalog of images, and helps the retailer enforce its compliance terms and conditions. We discuss in detail how we blend human expertise with state-of-the-art deep learning models to overcome a number of data and system level technical challenges outlined in Section 2.

The core learnings from this system can be utilized by any system that serves image or other visual content to human users on the web such as social media feeds, ads platforms, etc.

2. Technical Challenges

The proposed system is designed to address a number of data and system-level challenges as described below:

Figure 4. Challenge 3: Non-compliant category (e.g. Best Seller Badges) has various forms in which it can appear on images.

1. **Lack of Usable Training Data** : Most non-compliant images are hard to find. In most cases, the first example is discovered and reported by a customer. Even if we collect similar images from various sources on the internet, it is tens of labeled data points at best. Manual tagging is prohibitively expensive because the crowd needs to review thousands of images to find one true non-compliant example.
2. **Scale and Variation in Catalog**: E-commerce catalogs of large retailers have hundreds of millions of products, across tens of thousands of product categories. Additionally, the non-compliant content of any given type can appear across several, if not all, product categories. For example, the "best-price" logo can appear on images of products from any category. Moreover, the catalog data keeps changing. Creating a big enough training set that is a true representative of the catalog is difficult and expensive.
3. **Variety of Defining Examples**: A single non-compliant type can appear in multiple forms (e.g., a best seller badge) (Figure 4). We need to ensure that our models are generalized enough to capture various forms of infractions for a single use case.
4. **Custom and Fine-Grained Class Definitions** : The non-compliance guidelines often apply to a certain form or variation of an object. For example, most e-commerce websites allow hunting rifles but not assault rifles. From a machine learning point of view, differentiating between assault rifle and hunting rifle images falls into the category of fine-grained classifi-

cation which is challenging. Similarly, the image of a person wearing a swimwear is acceptable, but a picture of a nude person which is visually close to the former is not acceptable. This also means standard object detectors that detect guns or people would not suffice to solve our problem. An even more difficult manifestation of the problem is the case where certain images (such as a swimwear) are deemed offensive because of the pose or expression of the person, but other images featuring the same person are considered acceptable.

5. **Constraints on using text**: Even though each product comes with large amount of rich textual data, they are not easy to use for this problem. It is quite common for a compliant product to have a non-compliant image (e.g., a music CD with a nude photo on the CD) or vice versa. A title-based detector would fail to capture such an example. Alternately, optical character recognition (OCR) can be used to extract non-compliant text from the images alone. However, OCR works only if the image meets certain conditions. Also, OCR cannot capture a wide range of problems, such as nudity or an assault rifle, where there is no text on the image.

3. Related Work

The importance of images in e-commerce is well studied. Online shoppers often use images as the first level of information. Also, item popularity highly depends on the image quality [29]. [7] provides deeper understanding of the roles images play in e-commerce and shows evidence that better images can lead to an increase of buyers' attention, trust, and conversion rates.

Image classification models based on skin detection techniques [1] have been proposed for nudity detection. Skin regions are detected based on color, texture, contour, and shape information features. [30] uses maximum entropy distribution to detect the skin regions in the image.

Traditionally, logo/badge recognition has been addressed by keypoint-based detectors and descriptors [15, 23, 14], feature detection (using SIFT, SURF, BRIEF, ORB), and feature matching (using Brute-Force, FLANN matcher) [18] and classical template matching [19]. From our experience, these techniques do not work well for a catalog that contains millions of products. A few deep learning based logo detection models have been reported recently [3, 26, 9, 13, 10]. All of these techniques are tested on publicly available brand-logo datasets like BelgaLogos [14], FlickrLogos-32 dataset [24] or PL2K [10].

Recent advances in deep learning have brought neural nets to the forefront of image classification. A number of deep learning architectures such as Alexnet [16], VGG net [25], residual network [12], Inception [27, 28], and Nasnet [31] have been proposed. In this paper, we experimented with Resnet and Inception architectures that were

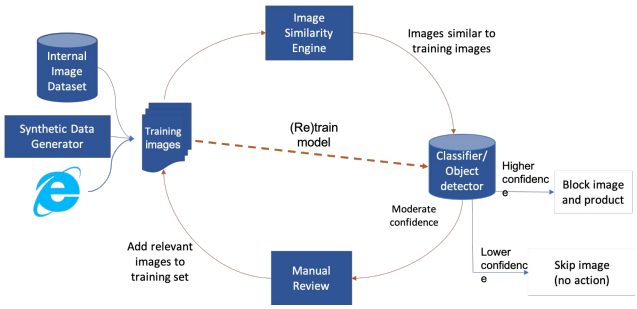

Figure 5. Proposed offensive and non-compliant image detection framework

pre-trained on an Imagenet [6] dataset, and then retrained on our images.

Object detection deals with detecting instances of semantic objects of a certain class and identifying the location of them. Some well-known object detectors are SSD[17], Region-based object detection[11], YOLO[21], R-FCN[5], and Faster R-CNN[22].

Generating synthetic training data allows for expanding plausibly ground-truth annotations without the need for exhaustive manual labelling. This strategy has been shown to be effective for training large CNN models, especially when sufficient training data is not available [8, 9].

4. Proposed Detection Framework

In this paper, we propose a computer vision powered framework, as outlined in Figure 5, for sparsely occurring content detection from images. In order to address the extreme scale, diversity, and dynamism of our dataset, we deviate from conventional approaches and innovate in a couple of ways.

1. **Iterative Training:** Unlike well-posed machine learning problems, we often start with a handful examples of offensive/non-compliant images. Hence, we collect data from various auxiliary sources and iterate a few times, as described in Section 4.1 to build training data.
2. **Transfer Learning:** It is impossible to train a neural net from scratch with the limited data we have. Hence, we leverage pre-trained networks and fine tune them with small but carefully crafted training data. Different training approaches are discussed in Section 4.2.
3. **Multi-stage Inference:** In order to scale, we propose to combine faster and lightweight classifiers with slower and deeper object detection networks. (Section 4.3)

4.1. Training Data Augmentation

Standard image data augmentation techniques such as translation, flip, rotation, color/contrast adjustment and noise incorporation are not sufficient for our application because we often start with a minuscule number of images.

We use the above mentioned controlled transformations, but we go beyond them and use additional novel techniques described below to solve the class imbalance problem

4.1.1 Visually Similar Image Search

As the first strategy, we leverage pre-indexed databases that are created to store signatures from millions of images and allow fast retrieval of similar images. The signatures are created from an Inception-v3 based deep learning model trained on all of catalog images for the purpose of product categorization. The embeddings from this model are re-used for various classification and retrieval tasks because they are generic representations of the deep latent factors of the image. In another variation, the signatures are created from VGG16 fc1 layer and then binarized to facilitate efficient indexing. Depending on number of images and the signature size, either FAISS or an ElasticSearch is used for indexing and approximate nearest neighbor search.

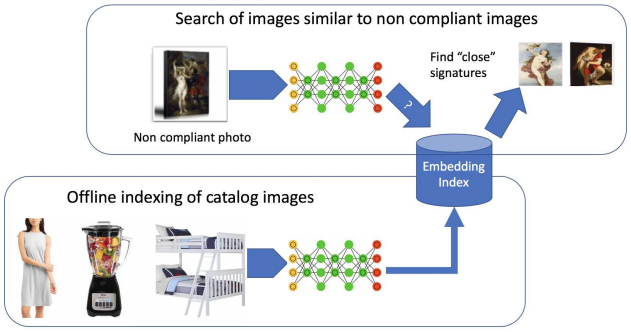

Figure 6. Building training data using visual search

As shown in Figure 6, for every training image, we compute its embedding using either model and then retrieve its nearest neighbors. We manually review the top few results and add some of them to the training set. Similar yet non-offensive images are added as valuable negative training data. For example, search with an image with nudity often retrieves underwear or lingerie model images which are not deemed offensive, but they serve as valuable training data.

4.1.2 Superimposition of Offensive Content

The above technique is effective for use cases where the entire object is prohibited such as assault rifle. However, we propose a different method for use cases like logos and badges where the problematic content is a very small part of the product image. Similar image search in such case would not work because the deep learning based signatures have more information about the main object in an image. For example, search by a hat with a certain brand logo will retrieve various hat images, not images of other products with the same brand.

Figure 7. Step a & c: Synthetic data generation using superimposition

Figure 8. Step d: Used training logo (top-left from figure 7) and applied random scaling, rotation and translation to generate a positive training sample as image (ii). Similarly, testing logo (top-right from figure 7) used to generate a test sample as image (iii). Image (iv) - Use similar-looking compliant logos collected in step b for superimposition.

We address this issue by generating synthetic training images in the following manner:

(a) We collect a large number of different-looking logos from the internet or from the data provider. We split the logo images into train and test sets (Figure 7).

(b) Not only non-compliant logos, we also collect images of similar-looking compliant logos whenever we have information about them. They will contribute to valuable negative examples. For example, confederate flag and Mississippi flag are quite similar looking.

(c) We tightly crop the logo images, leaving no space around and make the image transparent. (Figure 7)

(d) We apply controlled transformations on the logos, and then superimpose these logos on regular compliant images to make a non-compliant training or test sample. (Figure 8 - [ii] and [iii]). Compliant logos are used to create compliant training or test samples (Figure 8 - [iv]). Transformations include random scaling, rotating, orienting, flipping, translating, mangling, and/ or distorting the non-compliant content. (Figure 8 - [ii] and [iii])

Starting with approximately 100,000 compliant images sampled across the catalog representing all product categories, we apply the above mentioned steps to synthesize 100,000 positive samples for each type of non-compliant logo. Steps (b) and (d) help the model generalize better and reduce false positive rate. Since we know the exact location

of superimposition for every image, this process generates a large number of images with logos as well as accurate locations. Obtaining the bounding boxes at no cost is a big advantage of this synthetic data generation technique, as it dramatically reduces the cost of image annotation when training object detection models.

4.1.3 Crowdsourcing on Baseline Model Predictions

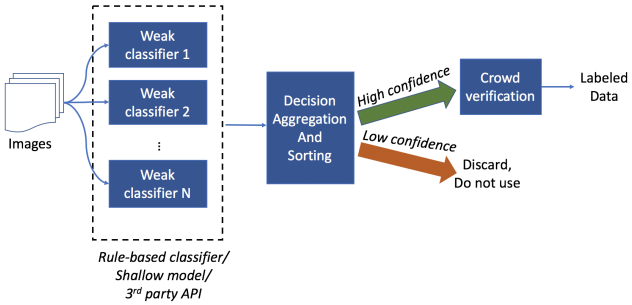

Figure 9. Manual verification of high confidence predictions from baseline predictions create better training data

We use the training data generated through the above mentioned processes to build shallow linear classifiers, small neural nets and heuristic based classifiers. In some case, we have access to commercially available classifiers from 3^{rd} party. All of these serve as baseline predictors that work as low precision and moderate recall. They are not nearly as good as the required level, however, we use them for a specific purpose (Figure 9). We run them on thousands of images from the catalog to generate predictions with confidence score. Depending on the available crowdsourcing budget, we decide on a confidence threshold. We send only the ones with confidence above that threshold to crowd or trained manual reviewers. They verify the baseline predictions and hence, generate high quality training data. Use of baseline predictors dramatically increases the return on investment for labeling because the high confidence predictions are more likely to be accurate.

4.2. Model Training

Depending on the amount of training data and the size and shape of the content to be detected, we employ one of the following three approaches.

4.2.1 Classifiers on Deep Embeddings

For problems like nudity and weapon detection, we build a classifier on top of image embeddings (Figure 10) extracted from a dense layer of the image similarity model mentioned in Section 4.1.1. This model is pre-trained on images from the entire product catalog. We experiment with Logistic Regression and Random Forest as classifier using a dataset of embeddings computed from the training set.

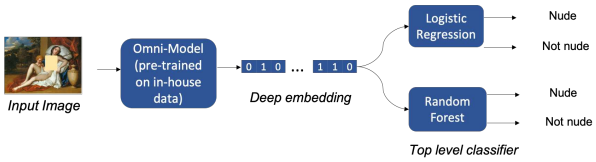

Figure 10. Approach 1: Classifiers on Deep Embeddings

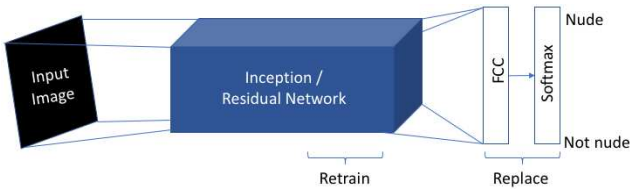

Figure 11. Approach 2: Retrain Last Layers of Deep CNN

4.2.2 Fine-Tuned Deep Neural Nets

For problems like logo where a pre-trained model is less likely to have learnt the concept or the object we need to detect, we retrain the last few layers of that pre-trained network with out data (Figure 11). We experiment with Resnet50 and Inception-V3, which were both pre-trained on Imagenet dataset. We remove the classification layer and add a fully connected layer and a softmax at the end. We vary the number of residual layers or the inception blocks to be retrained, to find optimal performance.

4.2.3 Object Detection

We use object detection for problems where fine-tuned classification networks do not perform well enough and we have images with annotated bounding boxes. We retrain Faster R-CNN to detect smaller objects such as logos, and we retrain YoloV3 to detect medium to large objects such as frontal nudity, sex toys or assault rifles. For YOLOV3, we run K-means clustering on the annotation boxes in the training data to determine a set of anchor boxes that represent the objects to be detected. The choice of K is domain and problem dependent. To give an example, we use $K = 2$ for nudity detection after carefully examining samples the training dataset. The upper and lower body nudity seems to have been captured by two different types of anchors.

Catalog images often have multiple figures/objects inside one image. Hence, during inference, both YOLOV3 and Faster R-CNN output one or many boxes with labels and confidence scores. For each image, we use the label of the box with highest confidence score as long as the confidence is above a threshold determined based on a hold-out set. The images labeled offensive are blocked or sent to human reviewers. Even if they flag the images as compliant, these images contribute as valuable training data. The object detector output is relatively more explainable than the classification methods because of the boxes.

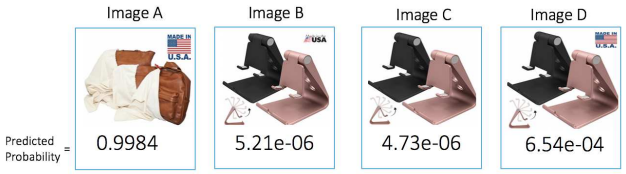

Figure 12. An example justifying the switch to object detectors from classification models.

Table 1. Approaches tried in different detection problems

Problem	Classifier on Embedding	Deep Neural Net	Object Detection
Nudity	Y	Y	Y*
Weapons	Y	Y*	N
Logo	N	Y	Y*

4.2.4 Selection of Training Method

A mix of intuition and data-driven insight drives the choice of technique for a given problem. To give an example, as we wanted to understand why the prediction from fine-tuned deep networks was wrong for a couple of logo test images (Figure 12), we first assumed that model is not generalizing well on different variants of the logo. In the example, image A and B both are non-compliant with different versions of *Made In USA* logo. While the model works perfectly fine on Image A, it was unable to detect Image B. To test our hypothesis, we created Image C which does not contain a logo at all, and image D that contains the exact same logo that was detected in A. The classification model could not detect the logo in image D, suggesting that the model was making decisions primarily by recognizing the main object and not the logo. Since it would be prohibitively costly to create a dataset comparable to the product catalog in terms of size and variety, we decided to switch to an object detector for the logo problem.

Table 1 presents which approach resulted in best performance for which problem. The ones tried are marked as Y, the best-performing one is marked with an asterisk.

4.3. Inference Strategy

In addition to model accuracy, two major driving factors of our system are time and compute resource cost. Running every image of the catalog through an array of deep learning models is prohibitively slow and costly. To address this issue, we make use of the observation that most non-compliance issues are more likely to appear within certain product categories. For example, nudity is most likely to be found in images of people, paintings, sculptures, CDs, carpets, books and posters. Assault rifle images are more likely to be found in hunting gears, toys, and books. This is why we use a broad image classifier as an entry-level filter (Figure 13). This first-level classifier (L1) classifies an input

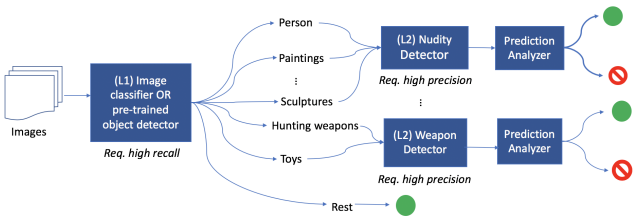

Figure 13. Two-stage classifiers for non-compliance issues(like nudity/weapons) that only occur in selected product categories. For non-compliance issues(like logos) which can occur in any product category, we use single-stage inference(not shown in diagram).

image into one of the major types, such as a person, book, painting etc. Depending on the type of image, it is send to one or more second-level detectors (L2) that are slower in inference and are trained to catch a particular non-compliant category. For example, an image of a person is expected to pass through the nudity detector, an image of a toy is expected to pass through the weapon detector, and so on. If an image does not fall into any of the product types associated with non-compliant categories, it is classified as 'rest' and it does not go through any L2 detectors. For non-compliance issues like best-seller logo which can occur on product images from any category, we use single-stage inference.

5. System Overview

The core models for nudity and weapon detectors are developed in Keras and served through Flask. The logo detector is based off a Tensorflow based implementation and served using Tensorflow-serving. To allow fast and reliable processing of hundreds of thousands of images everyday, the models are wrapped into microservices deployed through Docker. These microservices are integrated with the the overall image classification engine, as shown in Figure 14.

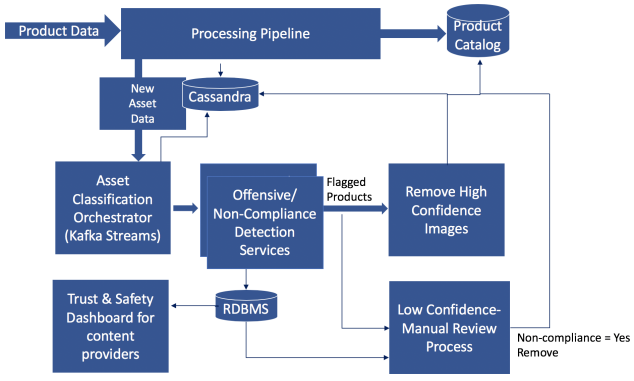

Figure 14. Overall System architecture

New product data including images is constantly fed to our e-commerce catalog by suppliers and sellers. The clas-

sification engine channels them to a Kafka queue. The queue is read by an orchestrator module that does some pre-processing such as size and format validation. Then, channels the image information to a number of queues dedicated to different detector micro-services. Each micro-service keeps reading from its own queue, processes the image with the model it hosts, and posts the results to a post-processing stream. Images that are flagged as non-compliant with high confidence are automatically removed from the catalog and the corresponding product is blocked. Images flagged with low confidence are pushed to a manual review pipeline. Based on the manual review budget, these images are reviewed in the priority assigned by the confidence score. Sellers and suppliers are given feedback through a dashboard that allows them to review and appeal their blocked content. This image classification system is designed to fit in a bigger product image selection system as in [4].

6. Results

In this section, we present all the experiments for "Best Seller" logos and nudity which represent non-compliant category and offensive category respectively. Experiments for other problems are similar in nature and have produced similar results and inferences.

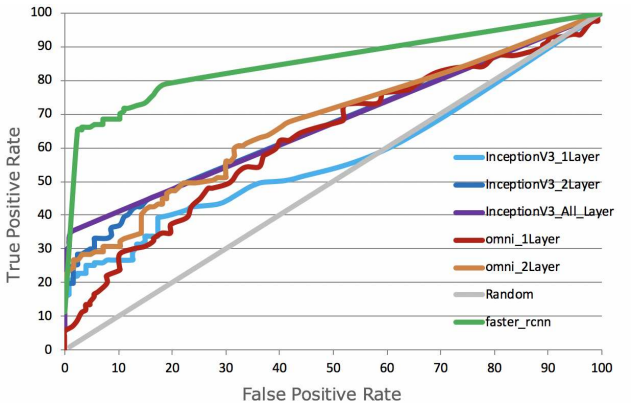

Figure 15. ROC curves based on Approach 2 and 3 for Logo Detection

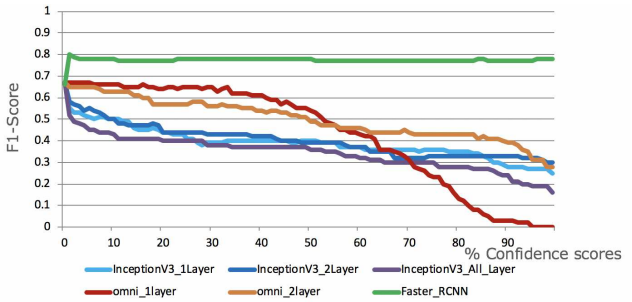

Figure 16. F1-Score for Logo Detection for various confidence thresholds

"**Best Seller" logos**: In this section, we compare the Faster R-CNN based detector that performs best against a number of other techniques. All the experiments were performed on a 700,000 train and 140,000 test set resized to 300X300. We first tried a number of baseline feature matching techniques such as SIFT and ORB feature descriptors followed by FLANN or BruteForce Matcher. We also tried multi-scale template matching. The results from these traditional techniques were not satisfactory, as shown in Table 2. The best f1-score is about 38%. The deep learning techniques performed much better, as shown in Figure 15 and 16. Linear classification of deep embeddings ((Section 4.2.1) is not applicable for logos. As for fine-tuned deep nets (Section 4.2.2), we retrained the last one, two, and all inception layers of a pre-trained InceptionV3. We also experimented with an in-house visual search model which is trained on the entire set of catalog images. We retrained its last one and two layers, the results for which are labelled as *omni_1layer* and *omni_2layer* in Figure 15 and 16. As seen in Figure 15, results from the InceptionV3 and the retrained visual search model are comparable to each other. For Faster R-CNN, we used Inception V2 as feature extraction net, pre-training on Coco dataset, momentum optimizer with initial learning rate of 0.0001 and IOU of 0.5. Figure 16 indicates that the f1-score of the Faster R-CNN model is 100% better than the retrained classification networks at a confidence score of 0.85.

We chose Faster R-CNN since it is known for delivering high accuracy on small objects such as logos. Faster R-CNN is one of the slower models among the popular object detection networks. Since our distributed architecture, designed based on queues allows higher inference time for image analysis, we consciously chose accuracy over inference time.

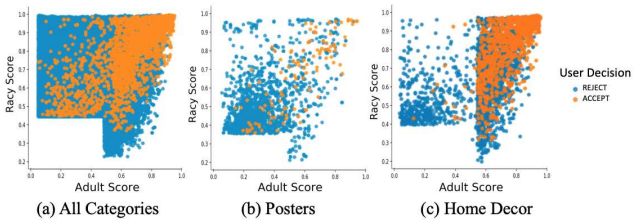

Figure 17. Manual Review Decision for images flagged by third party API for nudity detection

Nudity: Many commercial solutions for detecting nudity or sexually explicit content is available. Hence, we started one such third-party API that accepts an image and returns two scores, namely Adult Score and Racy Score, to quantify its offensiveness. The images with the two aforementioned scores above a certain threshold were sent for manual review to crowd workers. Since the third-party API is trained on different distributions of nude/sexual images compared

Table 2. Results for Baseline Logo Detector using traditional approaches

TECHNIQUE	PRECISION (%)	RECALL (%)	F1-SCORE (%)
SIFT + FLANN MATCHER	38.67	13.77	20.30
SIFT + BRUTEFORCE MATCHER	100.00	23.60	38.19
ORB + FLANN MATCHER	49.80	8.43	14.42
ORB + BRUTEFORCE MATCHER	47.71	4.17	7.66
MULTI-SCALE TEMPLATE MATCHING	45.55	4.43	8.08

Table 3. Results for Nudity Detector

TECHNIQUE	PRECISION (%)	RECALL (%)	F1-SCORE (%)
3RD PARTY API + MANUAL REVIEW	**X**	**X**	**X**
DEEP EMBEDDING + LINEAR CLASSIFIER (APPROACH 1)	+30	+34.5	+14
RESNET50 (APPROACH 2)	+32	+40	+35.5
INCEPTION-V3 (APPROACH 2)	+51	+49	+49.5
OBJECT DETECTION (APPROACH 3)	**+55**	**+67**	**+54**

to those in our catalog, the API returned a large number of false positives. As Figure 17a suggests, the percentage of images accepted by the crowd (denoted by orange dots) is far less than the count of those rejected by the crowd (denoted by blue dots). The FPR varies across categories (Figures 17b and 17c), but it is on the higher side regardless.

A month-long study of the manual review data revealed that (1) the presence of actual positive instances (nude images) was concentrated in a few segments of the catalog, (2) even within those categories, the distribution of positive and negative instances varied widely. Based on these observations, we fine-tuned the overall threshold and introduced category-specific thresholds. Even with all these changes, the best f1-score we could achieve was below 25%.

Nevertheless, the above baseline helped us create a larger training set for the deep learning approaches. Based on crowd responses for different categories, we built a training set that has enough representation of both positive and negative labels across all categories.

With the carefully crafted training and test data from baseline method, we experimented with three approaches shown in Table 3. This internal dataset contains about 5,000 positive training images with manually annotated bounding boxes and about 8000 test images, combining positive and negative classes. The goal of these experiments is to find a method that suits our use case/data, so absolute performance numbers are not presented. Instead, in Table 3, we compare a number of candidate techniques against the baseline method shown in bold with an **x**. The results from deep embedding based linear models (*Approach 1* from Section 4.2.1) and fine-tuned classification networks (*Approach 2* from Section 4.2.2) are much better than the baseline. Also, fine-tuned Inception v3 performs better than fine-tuned Resnet50. Since Approach 1 use signatures from a model trained on e-commerce catalog images and Approach 2 models are only pre-trained on Imagenet, the former technique generalizes better on new unseen images. Training the base model for Approach 1 is costlier though. Approach 2, which is based on a model trained on Imagenet, can be retrained faster with less data. In general, we observe that the quality and quantity of the data has a greater impact than the modelling choice. Finally, Approach 3 (YOLO v3 pre-trained on COCO and then fine-tuned) outperforms the rest. It achieves 54% lift in f1-score from the baseline.

We benchmark our nudity detector against the static images of **NPDI dataset**[2]. It contains 6387 / 10387 frames from porn / regular videos respectively. Since NPDI randomly selects frames from videos, we found about 15% of Porn-labeled frames had no nudity. Similarly, nonPorn frames contain bare-bodied males / females in casual settings. So, for a fair analysis of the model we had to remap Porn/non-Porn label to Nude/non-Nude in ecommerce context. Our model correctly classifies 98.8% frames as non-Nude and 95.38% as Nude.

7. Conclusion

In this paper, we present a computer vision powered system that detects and removes offensive and non-compliant images from an e-commerce catalog containing hundreds of millions of items. In addition to describing the core models of the system, we discuss the technical challenges of building a system at such a scale, namely, lack of training data, extreme class imbalance, and a changing test distribution. We also describe the critical refinements made to the data and to the modeling techniques to effectively overcome the challenges. This system is already deployed in production and it has processed millions of product images.

We plan to continue the work towards combining image and textual signals from products to build a more effective model. We are also trying to allow the system to detect unforeseen types of non-compliant cases with minimal amount of re-training and fine tuning of existing parameters.

The strategies adopted and the insights gained can be leveraged by content-serving web-based platforms from other domains as well.

References

[1] W. A. Arentz and B. Olstad. Classifying offensive sites based on image content. *Computer Vision and Image Understanding*, 94(1-3):295–310, 2004. 3

[2] S. Avila, N. Thome, M. Cord, E. Valle, and A. D. A. AraúJo. Pooling in image representation: The visual codeword point of view. *Computer Vision and Image Understanding*, 117(5):453–465, 2013. 8

[3] S. Bianco, M. Buzzelli, D. Mazzini, and R. Schettini. Deep learning for logo recognition. *Neurocomputing*, 245:23–30, 2017. 3

[4] A. Chaudhuri, P. Messina, S. Kokkula, A. Subramanian, A. Krishnan, S. Gandhi, A. Magnani, and V. Kandaswamy. A smart system for selection of optimal product images in e-commerce. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 1728–1736. IEEE, 2018. 7

[5] J. Dai, Y. Li, K. He, and J. Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, pages 379–387, 2016. 3

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009. 3

[7] W. Di, N. Sundaresan, R. Piramuthu, and A. Bhardwaj. Is a picture really worth a thousand words?: - on the role of images in e-commerce. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, WSDM '14, pages 633–642, New York, NY, USA, 2014. ACM. 3

[8] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015. 3

[9] C. Eggert, A. Winschel, and R. Lienhart. On the benefit of synthetic data for company logo detection. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 1283–1286. ACM, 2015. 3

[10] I. Fehérvári and S. Appalaraju. Scalable logo recognition using proxies. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 715–725, 2019. 3

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Region-based convolutional networks for accurate object detection and segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):142–158, 2016. 3

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[13] F. N. Iandola, A. Shen, P. Gao, and K. Keutzer. Deeplogo: Hitting logo recognition with the deep neural network hammer. *arXiv preprint arXiv:1510.02131*, 2015. 3

[14] A. Joly and O. Buisson. Logo retrieval with a contrario visual query expansion. In *Proceedings of the 17th ACM International Conf. on Multimedia*, pages 581–584, 2009. 3

[15] Y. Kalantidis, L. G. Pueyo, M. Trevisiol, R. van Zwol, and Y. Avrithis. Scalable triangulation-based logo recognition. In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, page 20. ACM, 2011. 3

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. 3

[17] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 3

[18] A. Mordvintsev and A. K. Feature matching. https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_matcher/py_matcher.html, 2013. 3

[19] A. Mordvintsev and A. K. Template matching. https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_template_matching/py_template_matching.html, 2013. 3

[20] J. Nielsen. How users read on the web. https://www.nngroup.com/articles/how-users-read-on-the-web/, 1997. 1

[21] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 3

[22] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 3

[23] S. Romberg and R. Lienhart. Bundle min-hashing for logo recognition. In *Proceedings of the 3rd ACM conference on International conference on multimedia retrieval*, pages 113–120. ACM, 2013. 3

[24] S. Romberg, L. G. Pueyo, R. Lienhart, and R. Van Zwol. Scalable logo recognition in real-world images. In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, page 25. ACM, 2011. 3

[25] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3

[26] H. Su, X. Zhu, and S. Gong. Deep learning logo detection with data expansion by synthesising context. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 530–539. IEEE, 2017. 3

[27] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014. 3

[28] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015. 3

[29] S. Zakrewsky, K. Aryafar, and A. Shokoufandeh. Item popularity prediction in e-commerce using image quality feature vectors. *CoRR*, abs/1605.03663, 2016. 3

[30] H. Zheng, H. Liu, and M. Daoudi. Blocking objectionable images: adult images and harmful symbols. In *2004 IEEE International Conference on Multimedia and Expo (ICME)(IEEE Cat. No. 04TH8763)*, volume 2, pages 1223–1226. IEEE, 2004. 3

[31] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le. Learning transferable architectures for scalable image recognition. *CoRR*, abs/1707.07012, 2017. 3