Automated Transformation of Photoshoot Images into Promotional Banners at Scale

Shivang Singhal
shivang.s@flipkart.com
Flipkart Internet Pvt. Ltd.
Bangalore, India

Abhay Gupta
abhay.g@flipkart.com
Flipkart Internet Pvt. Ltd.
Bangalore, India

Aditya Sharma
aditya.ss@flipkart.com
Flipkart Internet Pvt. Ltd.
Bangalore, India

Aditya Ramana Rachakonda
aditya.rachakonda@flipkart.com
Flipkart Internet Pvt. Ltd.
Bangalore, India

ABSTRACT

In an e-commerce marketplace, creating visually rich and personalised content for the home page is a key component in driving engagement and purchases. Brands tend to have a lot of high quality photoshoot images and e-commerce marketplaces tend to have good recommender systems for their catalogue of products. Being able to create visually rich content from pre-existing photoshoot images and mapping them to content in the catalogue is currently a manual process which does not scale. This paper delves into a method, termed *photo transformer*, consisting of a sequence of steps by which we create personalised content from generic but rich photoshoot images. In our testing we found that we can significantly improve the user engagement through this process.

ACM Reference Format:
Shivang Singhal, Abhay Gupta, Aditya Sharma, and Aditya Ramana Rachakonda. 2022. Automated Transformation of Photoshoot Images into Promotional Banners at Scale. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

In an e-commerce marketplace, participating brands like to reach their audiences using visually rich and attractive banner images with highly personalised offers, and corresponding call-out texts. Recommender systems have seen significant advances in the last decade and currently they are really good at personalising content and conveying the relevance of the content to the user. The one stumbling block has been to present them in a visually rich manner so as to make the content seem less like grids of data coming from a database and more like catchy rich content present in magazines.

Brands spend several millions of dollars every year generating high quality photoshoot images but these tend to be for the print medium and cannot be easily personalised using a recommender system onto a mobile application of an e-commerce marketplace. The process of converting them to be used on the mobile to sell content is highly manual as it involves subjective decisions and aesthetics of banner designers.

The goal of this paper is to take a large set of such photoshoot images from brands and automatically generate visually rich and personalisable banners at scale with no human interventions. This allows brands to upload a large set of high-quality images and these would be redesigned and used by the e-commerce marketplace to promote content of that brand automatically using pre-existing recommender systems.

In this paper we discuss *photo transformer*, the method we used to convert a photoshoot image into a usable banner. We restrict our discussion to fashion but this is primarily a limitation of our testing and not necessarily a limitation of the techniques. We do this by identifying the humans in the image and then their apparel. We then find ways to create a banner using that information and then tag the image with these attributes for the recommender systems to utilise the banners. Figure 1 shows the transformation of one such photoshoot image into a banner. In this paper, for reasons of copyright and anonymity we refrain from using brand logos and highly personalised call-out texts. So in production, the content would look more brand-centric and lot more personalised to users.

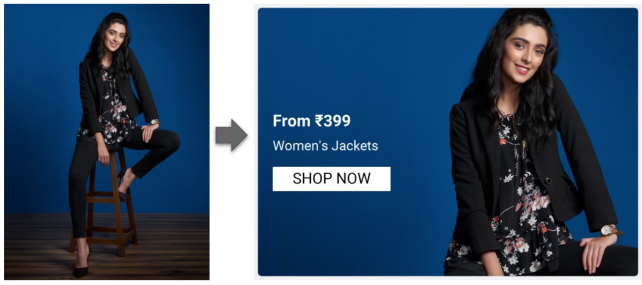

Figure 1: Photo transformer

2 RELATED LITERATURE

Photo transformer rely heavily on methodologies from computer vision and is primarily based on object detection. Object detection [1] is a computer vision technique that allows us to identify and locate semantic objects in an image or video. Deep learning and Convolution Neural Networks (CNN) [2] have changed the landscape

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

of object detection in terms of faster and accurate architectures. Modern deep learning based object detectors [3] are broadly classified into two categories - "two-stage detectors" and "one-stage detectors". They differ in terms of speed and accuracy. Two stage detectors are usually more accurate while one stage detectors have faster inference time.

In two stage detectors, the first stage is called region proposal network (RPN); it takes in an input image and outputs region proposals, which are locations where an object might be present and the second stage involves classifying to which class each proposal belongs to and refining the bounding boxes. In R-CNN [4] RPN runs through original image and proposes regions where object might be present. These proposal regions are warped to same size and passed through a CNN and then SVM to classify class and a regressor to predict exact bounding box. In Fast R-CNN [5] RPN runs through the ConvNet feature map which is output of CNN, rather than the image.

Faster R-CNN [6] is successor of Fast R-CNN, and here RPN is the convolution neural network itself and is part of the whole network. To generate region proposals, RPN slides a small network over the convolutional feature map output by the last shared convolutional layer.

More recent advancements like Feature Pyramid Networks [7] (FPN) and Cascade R-CNN [8] build upon Faster R-CNN and improve the accuracy and speed of the algorithm.

In one stage object detectors like Single Shot Detector (SSD) [9], instead of having another neural network (or any algorithm) to generate region proposals, an entire grid of feature map is considered as region proposals which in turn is classified by the neural network to produce class scores and bounding box offsets. SSD turns out to be 5-6 times faster during inference time than region proposal based networks while maintaining competitive accuracy. SSD suffers from very high class imbalance between positive classes (an object of relevance) and negative (background) class. The classifier gets more negative samples compared to positive samples, thereby causing biased learning. To combat extreme foreground-background class imbalance encountered during training, a loss function called "focal loss" is used. It reshapes the standard cross entropy loss so that detector will put more focus on hard, misclassified examples during training.

Most recently, EfficientDet [10], which uses a weighted bi-directional feature pyramid network (BiFPN) and EfficientNet [11] as a backbone is shown to outperform all previous algorithms.

Applications similar to photo transformer had been attempted by companies like Netflix and Myntra. On the home page, Netflix [12] personalises cover-art of movies, TV shows and other content based on the user's interests. It changes look and feel of the same content for different users by understanding what might be the most interesting aspect of that content for them. Myntra [13] has experimented with genetic algorithms and a heuristic driven energy function to place the brand logo and call out text on a photoshoot image. The entire pipeline is quite similar at a high level but different in the internals. While objective for them is to place brand provided text as well logo on the given image appropriately.

3 PHOTO TRANSFORMER

Photo transformer automates content creation by utilising photoshoot images and generates presentable banners at scale. Brands regularly conduct photoshoots of various products for promotional purposes and these are vibrant, stylish and brand agnostic images. They look quite different from our catalogue images which are focused on showing only the product on an uninteresting but consistent white background. Utilizing more of photoshoot images can also make an e-commerce homepage look more diverse and visually appealing.

Content personalisation happens by personalising textual call outs to the user and by matching the right products to the right users in the ranking layers. We also personalise a content by placing different call out texts depending on the user's interest. For example if user has an affinity towards lower pricing, the call out could be about budget jeans and for a different user it could be about premium jeans. Based on the call out we dynamically set the right landing page for that call out.

3.1 Components

We have built a pipeline of models (see figure 2) and algorithms to morph images from a photoshoot into a suitable banner with a marketable product collection. First, we detect the persons in the image along with their gender. We then detect various body parts like hands, legs and face and also facial features like eyes and nose. In parallel we also detect the items of clothing or accessories that are in the image. In all these detections we identify the bounding box along with the appropriate label. We then crop the image according to the best possible fit from a banner creation standpoint. We then extend the image to fit the aspect ratio of the final banner that needs to be displayed. If the background does not have sufficient contrast to write text, we fade the background into a gradient which contrasts with the text and which goes well with the background. The call out text is then placed to complete the banner (see figure 3).

3.1.1 Apparel Detector. This is an object detection model built to detect apparel in an image. This model detects a bounding box along with its class. To train an object detection model, we require a dataset of images annotated with bounding boxes and the corresponding contained class of object. Our org's catalogue images only contained class information and no bounding boxes.

We identified relevant classes (see figure 4) from the Open Images Dataset (OID) [14] and used this data to train object detection model. Training model using OID data was also challenging due to the variety of images it contained and the annotations though detailed, were noisy e.g. there are missing annotations [15], there is a huge class imbalance between classes etc.

On our platform we also sell products which represent Indian ethnic wear like those shown in figure 5 and these categories are not present in OID dataset. We also trained an object detection model for those categories using our own catalogue images. However it was challenging because of the absence of bounding box information in catalogue images and required manual labelling. Manual tagging for each image which would have been time consuming and is explosive with the number of classes. We devised an approach to train our object detection model to detect these classes as well without any manual tagging, explained below.

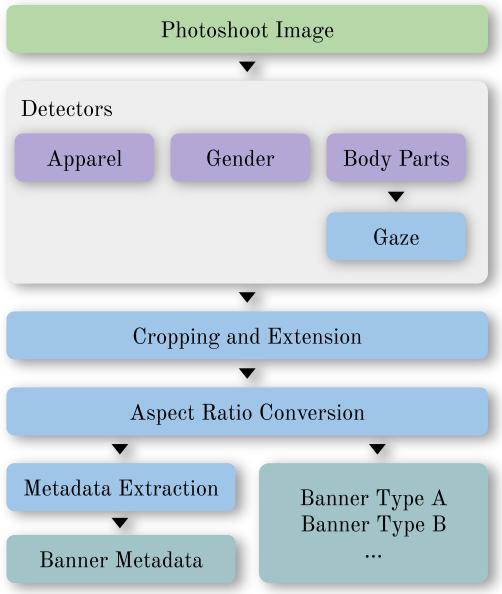

Figure 2: Photo transformer architecture

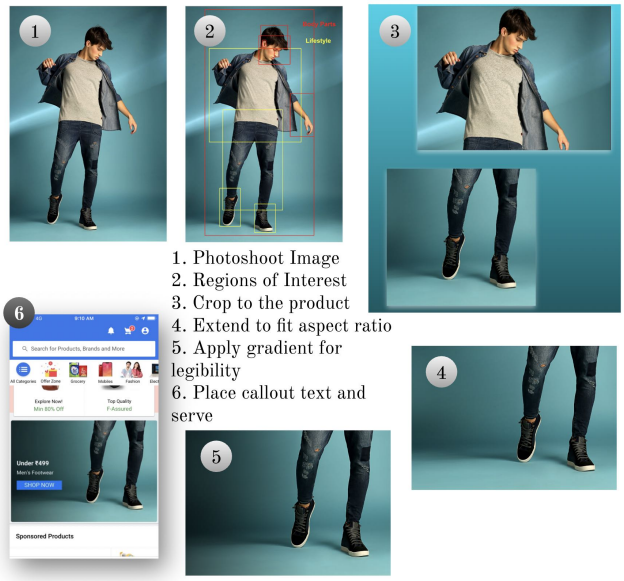

Figure 3: Photo transformer pipeline

Belt	Sunglasses	Backpack	Hat	Skirt
Shorts	Dress	Swimwear	High heels	Trousers
Shirt	Earrings	Coat	Jacket	Handbag
Boot	Necklace	Suit	Sandal	Footwear

Figure 4: Global apparel classes

Figure 5: Indian ethnic wear: kurta, lehenga, saree, legging

Men	Dhoti	Kurta	Sherwani	Ethnic Set
Women	Churidar	Bottom wear	Legging	Ethnic Set
	Kurta	Kurti	Lehenga	Patiala
	Salwar	Dupatta	Saree	Ethnic skirt

Figure 6: Indian ethnic wear classes

Utilising OID data we trained an object detection model to detect three class of clothing: upper body clothing, lower body clothing and full body clothing. We also categorised Indian wear into three categories for example *Saree* and *Lehenga* is full body clothing, *Kurta* is upper body clothing, *Leggings* as lower body clothing. The catalogue images were then passed through this model which gave us bounding boxes for the three generic classes. Since we already knew the finer class of the image, we picked the bounding box selectively leaving out the box which does not belong to image's parent class. After cleaning the resultant data we chose as comprehensive set of Indian ethnic wear apparel types (see figure 6). We trained an object detection model to detect both Global and Indian apparel. We experimented with multiple models and this is summarised in Table 1.

Quantitative results show model performances on the test dataset which is held out from the same dataset the model was trained on. Our primary metric is mean average precision (mAP). By mAP we mean precision computed where intersection over union (IoU) greater than 0.5. This is typically notated as mAP@0.5. For recall we used mean average recall where we look at the recall when we choose 10 items (average recall at 10 or AR@10). These are in line with the COCO challenge [16].

Even though we were able to achieve reasonable model performance on held out data, performance of the model on photoshoot images was not comparable to that of the held out set. Photoshoot images can be visually very different from training data and there are instances where clothing item present in photoshoot looks pretty different from its representation in training data as clothing item present in photoshoot might be more trendy, stylish and posing is different. Hence qualitative evaluation of model on photoshoot image is also required. We identified a set of challenging photoshoot images and initially we used to visualise model detections on these images to identify the best model. To make this process more systematic we tagged items present in these images and now we measure same metrics i.e. mAP and mAR on these images. We term these as qualitative metrics present in table 1.

Typical evaluation process for object detection can be summarised as identifying candidate checkpoints from a metric (mAP, mAR) progression with respect to training steps. For example, finding best performing model on test data and few other models which might have lesser performance but also took lesser training steps to train and hence they might be able to generalise well on photoshoots, we call them candidate checkpoints. We then do qualitative evaluation of these candidate checkpoints on photoshoot images and pick the best one.

We experimented with both single stage (YOLOv4 [17], EfficientDet) and two stage (Faster R-CNN) architectures. Faster R-CNN Lite is Faster R-CNN model with Resnet 101 as backbone while Faster R-CNN heavy is Faster R-CNN with Inception Resnet v2, Atrous version. Even state of the art model EfficientDet has lesser precision than best performing Faster R-CNN architecture and performance difference is more visible in qualititative evaluation, its recall is notably better than all the models though. The two stage architectures are inherently better at performance and that explaines why YOLOv4 is lagging behind but EfficientDet is shown to outperform previous architectures and currently considered as state of the art. Our assumption behind its low performance is it could be because of missing annotations in the data and Faster R-CNN architecture by its design is least impacted by missing annotation issues [15]. Since our primary goal is to have model with highest precision, we used Faster R-CNN Heavy for apparel detection.

Algorithms	Quantitative		Qualitative	
	mAP	mAR	mAP	mAR
Faster R-CNN Lite	60.3	56.5	35.6	30.9
Faster R-CNN Heavy	**62.6**	66.0	**47.9**	46.6
YOLOv4	57.0	50.2	20.7	35.4
EfficientDet	61.9	**71.3**	43.4	**53.9**

Table 1: Object Detection Experiments

Table 2 also lists incremental steps we took to improve object detector performance and relative performance gain achieved after every step in the apparel detector.

3.1.2 Gender and Body Parts Detector. We created a dataset of human body parts and gender utilizing OID. We trained the model to detect various classes, namely boy, girl, beard, body, ear, eye, face, feet, hair, hand, head, leg, mouth, man and woman. This model was used to detect the gender and location of various body parts of the human in the photoshoot image.

Here we dropped using single stage detectors and only experimented with both versions of Faster R-CNN. Best performing model came from Faster R-CNN model with ResNet-101 as the backbone i.e. Faster R-CNN lite. This model was robust on photoshoot images and it generalised better on photoshoot images as compared to apparel detector models. This is expected because humans present in photoshoot images will be no different than their representation in training data. We followed the same methodology and selected a model checkpoint which was doing relatively better on photoshoots by checking visually on sample images. Quantitative numbers of models in our experiments with are reported in table 3

Steps	mAP Progression
ImageNet head, including body parts classes	22.0
Change head to COCO ResNet-101	33.2
Separate Human Body Parts and combine visually similar classes	41.6
Data augmentation for selected classes	46.0
Double input size and more data augmentation	50.0
Downsample/Remove few majority classes, remove single annotation files	55.0
Hyperparameter tuning, Classification vs Regression loss weight change, IoU threshold variation, Optimizer experimentation, Anchor Box aspect ratio identification using data etc.	**60.3**

Table 2: Object Detector Training Steps

Model	mAP	mAR
Faster R-CNN Lite	37.9	46.9
Faster R-CNN Heavy	37.14	46.6

Table 3: Gender and Body Parts Detector

3.1.3 Gaze Detector. Based on feedback from our design team, we learnt that a good banner is one where the person in the image does not look away from the text. So we wrote a heuristic which can categorise the direction of gaze into `center`, `left` or `right` by observing the relative positions of the facial features.

If nose and mouth are detected then we observe if they are significantly to the right or left of the face to identify the direction of gaze. When they are not detected we use the position of the body relative to the face to approximate the gaze. Gaze detection module accuracy is 97.3%. An example is shown in figure 7.

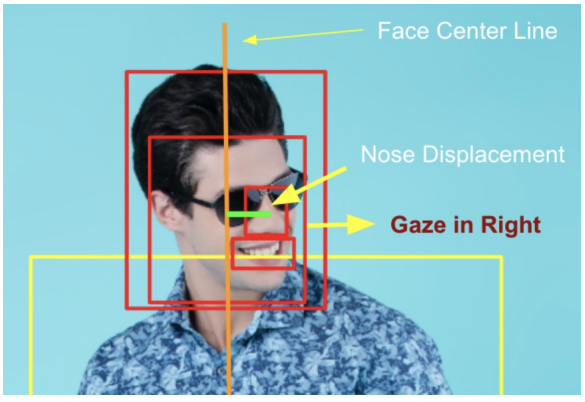

Figure 7: Gaze detection

3.1.4 Cropping and Extension. This module generates candidate banner images after appropriately cropping apparel and extending

the background to maximum possible width. This module makes sure that the images generated look good and are free from inconsistencies. Cropping the right way is dependent on a large number of heuristics which guide the sense of style.

Class-wise cropping rules are specific to each apparel class, for example, cropping jeans requires the presence of footwear, cropping upper body clothing requires the presence of a face. Another example is the way we crop footwear as opposed to say, shorts. Footwear need to be paired and the pair should be clearly present in the image.

We also crop two classes together if they are very close to each other. For example, two people standing next to each other wearing t-shirts. Items should have some breathing space adjacent to them so that they are not pushed to corners when cropped.

The direction of the gaze determines the area to place the text and the crop should extend to as much of the original image as possible in this direction. This is done while ensuring we do not include other classes or body parts.

3.1.5 Aspect Ratio Converter. Banners are served in multiple design formats on our e-commerce platform while their design is guided by template definitions. From content creation perspective the width and the height of the final image are most relevant. Also whether the text would be rendered on the image or below the image affects the crop as the object needs to be aligned in the centre if the text is below. If the image falls short of the aspect ratio in width, we expand it automatically by stretching pixels which do not contain any information regarding the foreground object classes. This module generates final banner images over which a callout can be placed, banner image is created maintaining multiple conditions, such as apparel must be properly visible, there is sufficient space to write text and template aspect ratio is maintained. Next the generated banner image is resized to exact width and height of the template.

3.1.6 Metadata Extractor. Along with the final image we also generate metadata which provides the landing page store information based on the class of the item and the gender of the person in the image. We also provide a colour palette which works best with the background and which is computed based on the luminosity of the image. This colour palette helps determining the callout text colour, we place complementary but within business defined text colour range so that text is easily readable for the user.

Photoshoot images may contain busy scenes in background, like a plant or a textured wall. These make it hard to overlay a legible call out text. In such cases, we apply a gradient on the image so as to improve legibility without hindering the visual quality of the image.

To identify such noisy backgrounds, we compute the channel-wise pixel variation and use a well-known heuristic [18] to get the noise coefficient.

$var_{luma} = .299^2 * r_{std} + .587^2 * g_{std} + .114^2 * b_{std}$

r_{std} represents standard deviation of R channel, g_{std} represents standard deviation of G channel, b_{std} represents standard deviation of B channel in RGB image. var_{luma} is noise coefficient and when $var_{luma} > 8$ we apply a gradient. Example of gradient operation is shown in figure 8 .

After all these steps a banner is created and after placing relevant text and attaching landing url to each banner its served on the homepage. Few sample banners which we served are shown in figure 9. These banners are prior to placing user personalised text. Banners contain all the necessary information e.g. gender, store and background space coordinates to identify where to write text along with it's colour.

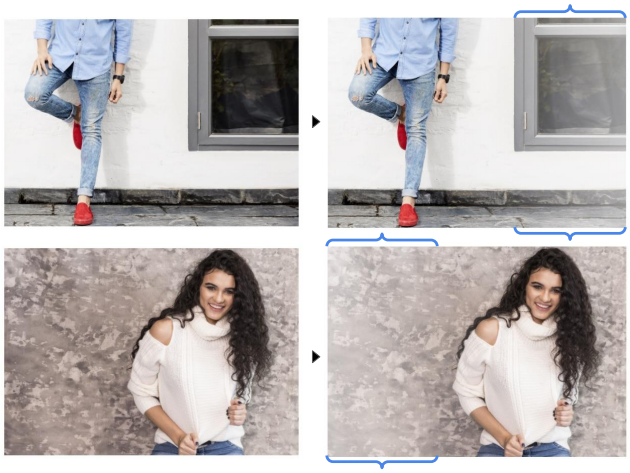

Figure 8: Subtle fade out gradient

4 PERFORMANCE AND BUSINESS IMPACT

Photo transformer generated banners are being served on our homepage regularly. The prime driver of business performance is engagement which we measure using click-through rate (CTR). It is important to note that clicks on the homepage are motivated mainly by the creative banner displayed, while the conversion is driven by the quality of the collection shown at the landing page.

In our experiment we controlled for size and position by showing manually created content and photo transformed content of the same size at the same slot. We found that the CTR of photo transformer generated banners were consistently better and overall were 2.15 times greater than static manual content.

Figure 10 shows comparison between manually created content and photo transformer generated content over a period of 4 weeks. It can be inferred that Photo Transformer generated content has been doing much better than manual content, consistently. The CTR increased because photo transformer generated banners are highly user personalised while most of manual generated visually rich content is non-personalised and generic.

5 CONCLUSIONS

We have created an automated system to generate banners from photoshoot images at scale. We found that these significantly outperform manually created banners. In the future, we plan to extend this work to other categories which are photoshoot dependent like furniture, furnishings, jewellery and so on.

REFERENCES

[1] Z. Zou, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *arXiv preprint arXiv:1905.05055*, 2019.

Male; Suit; Text on Right in Black | Male; Shoes; Text on Right in Black | Male; Shirt; Text on Left in Black

Female; Saree; Text on Left in Black | Female; Dress; Text on Right in White | Male; Trousers; Text on Right in White

Figure 9: Sample banners with machine-generated tags before adding personalised call-out text

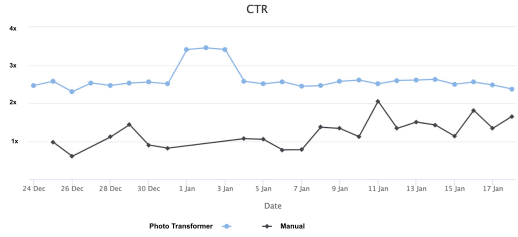

Figure 10: Photo transformer vs manual CTR

[2] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, 2017, pp. 1–6.

[3] L. Jiao, F. Zhang, F. Liu, S. Yang, L. Li, Z. Feng, and R. Qu, "A survey of deep learning-based object detection," *IEEE Access*, vol. 7, pp. 128 837–128 868, 2019.

[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[5] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, pp. 91–99, 2015.

[7] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[8] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6154–6162.

[9] 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*. Springer, 2016, pp. 21–37.

[10] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov *et al.*, "The open images dataset v4," *International Journal of Computer Vision*, pp. 1–26, 2020.

[11] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 6105–6114.

[12] J. B. Ashok Chandrashekar, Fernando Amat and T. Jebara, "Artwork personalization at netflix," https://netflixtechblog.com/artwork-personalization-c589f074ad76, Dec. 2017.

[13] S. Vempati, K. T. Malayil, V. Sruthi, and R. Sandeep, "Enabling hyper-personalisation: Automated ad creative generation and ranking for fashion e-commerce," in *Fashion Recommender Systems*, 2020, pp. 25–48.

[14] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, T. Duerig, and V. Ferrari, "The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale," *IJCV*, 2020.

[15] Z. Wu, N. Bodla, B. Singh, M. Najibi, R. Chellappa, and L. S. Davis, "Soft sampling for robust object detection," *arXiv preprint arXiv:1806.06986*, 2018.

[16] "Coco detection metrics," https://cocodataset.org/#detection-eval.

[17] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[18] O. de Lama, "How to compute rgb image standard deviation from channels statistics," https://www.odelama.com/data-analysis/How-to-Compute-RGB-Image-Standard-Deviation-from-Channels-Statistics/.