Unposed: Unsupervised Pose Estimation based Product Image Recommendations

Saurabh Sharma
Amazon.com
Bengaluru, India
sharsar@amazon.com

Faizan Ahemad
Amazon.com
Bengaluru, India
ahemf@amazon.com

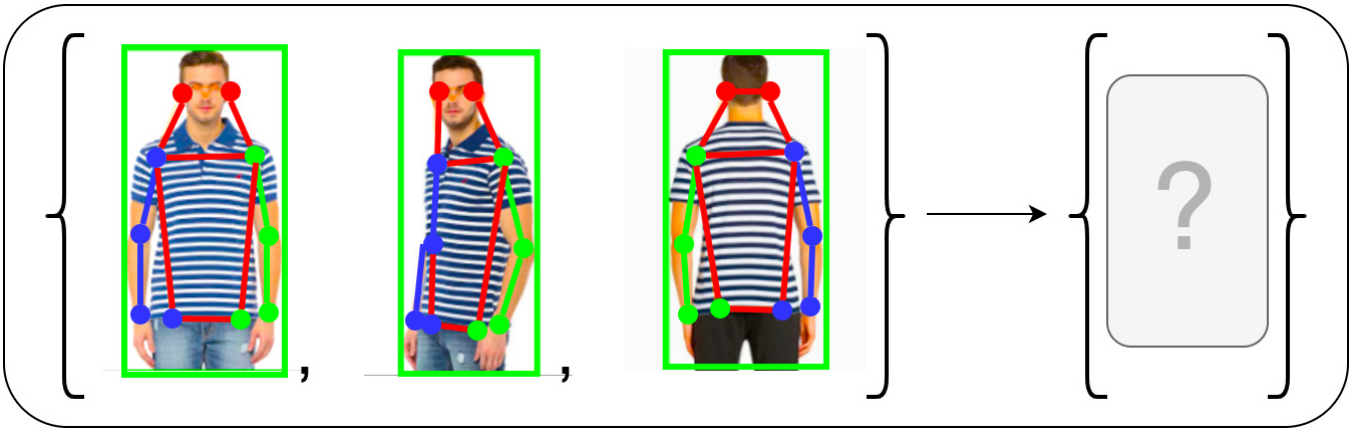

Figure 1: Pose detection on Product images to propose images.

ABSTRACT

Product images are the most impressing medium of customer interaction on the product detail pages of e-commerce websites. Millions of products are onboarded on to webstore catalogues daily and maintaining a high quality bar for a product's set of images is a problem at scale. Grouping products by categories, clothing is a very high volume and high velocity category and thus deserves its own attention. Given the scale it is challenging to monitor the completeness of image set, which adequately details the product for the consumers, which in turn often leads to a poor customer experience and thus customer drop off.

To supervise the quality and completeness of the images in the product pages for these product types and suggest improvements, we propose a Human Pose Detection based unsupervised method to scan the image set of a product for the missing ones. The unsupervised approach suggests a fair approach to sellers based on product and category irrespective of any biases. We first create a reference image set of popular products with wholesome imageset. Then we create clusters of images to label most desirable poses to form the classes for the reference set from these ideal products set. Further, for all test products we scan the images for all desired pose classes w.r.t. reference set poses, determine the missing ones and sort them in the order of potential impact. These missing poses can further be used by the sellers to add enriched product listing image. We gathered data from popular online webstore and surveyed ~200 products manually, a large fraction of which had at least 1 repeated image or missing variant, and sampled 3K products(~20K images) of which a significant proportion had scope for adding many image variants as compared to high rated products which had more than double image variants, indicating that our model can potentially be used on a large scale.

AIMLSystems 2022, October 12–15, 2022, Bangalore, India

ACM ISBN 978-1-4503-9847-3/22/10…$15.00
https://doi.org/10.1145/3564121.3564126

CCS CONCEPTS

• **Computing methodologies** → *Shape inference.*

KEYWORDS

pose detection, interpretable, image tagging

ACM Reference Format:
Saurabh Sharma and Faizan Ahemad. 2022. Unposed: Unsupervised Pose Estimation based Product Image Recommendations. In *The Second International Conference on AI-ML Systems (AIMLSystems 2022), October 12–15, 2022, Bangalore, India.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3564121.3564126

1 INTRODUCTION

On the e-commerce stores, the product information is presented to the customer primarily in the 3 modes: *text, video and image.* An accurate and wholesome presentation of the product details is critical to influence purchase decision of the customer. With more and more sellers offering an even wider selection of products, it

becomes unscalable to maintain the catalogue quality by checking for missing information.

Depending on the domain of the business the 3 modes of presenting product details may have different impact. For example, a customer purchasing an electrical component may be more interested in detailed specifications in text format, whereas the one, a winter jacket, may be interested in visual image of the product. Even for image friendly categories, given the variety and nuances of products, a one-size-fits-all solution for absent images is ineffective. So we focus on fast moving Apparel product type and outline a solution to suggest missing images in the following steps:

(1) Determine ideal imageset in an unsupervised manner.
(2) Scan all products of the category for missing images against this set.
(3) Propose the image candidates ordered by potential impact.

1.1 Catalogues in clothing

Clothing product pages tend to show images of human models / mannequins exhibiting the clothing / accessories from different poses / directions. However, many products on the webstores may suffer from poor cataloguing practices like duplicated images, bad image lighting, poor background, insufficient number of angles of clothing, etc. leading to poor customer experience. A few A/B experiments done at Amazon have confirmed this and have shown a non-trivial positive impact of having larger number and varied type of images.

The problem of inadequately imaged product, thus has a user experience improvement incentive. Any automation that can meaningfully nudge the selling partners with right suggestions for improving their catalogue images will help them better their listing. This in turn can improve the end user experience, there by driving the virtuous cycle.

The missing images can be indicated to the sellers in text hints, or they can be provided reference image set of popular products in the Category/ Subcategory corresponding to their underrepresented product in the clothing category. For example, Figure 2 shows the *reference* imageset with 6 images, covering *front, farview, neckview, back, fabric pattern, side view* etc. whereas, the subject product image are missing a few from the reference set. In this work, we discuss ways to detect, and indicate the missing images to improve the product listing by detecting missing poses in the image set.

2 RELATED WORK

This section discusses prior work done for catalogue quality improvement as well as gives a brief refresher on the problem of human pose detection.

2.1 Catalogue Image quality

The product listing image selection is largely done manually by professionals specializing in product listing adhering to Standard Operating Procdeures based on listing guidelines. Some prior published work [4] suggests solutions to build systems to solve image listing problems like de-duplication, image cropping etc.

In our scheme we suggest what images can be added to the Product imageset to enhance the user experience, and no prior work in our knowledge discusses pose detection (discussed next §2.2) as a method to generate candidates for image addition.

2.2 Pose Detection

Human pose detection has been a widely studied computer vision application. The core problem of human pose detection is to identify the body landmarks or regions/points of interests (PoI) like *nose, shoulder, knee* etc. and predict coordinates of these landmarks *w.r.t.* the given image. A few prominent models use HourGlass[8] and U-NET[10] type of Encoder-Decoder NN architecture to identify Points of Interest.

Then there are multistage approaches where in the Bottoms-Up approach the model first detects body joints location on the image and then links them to guess complete human instance. In Top-down approach models [7], in the first stage the model identifies the human body in the image, and in subsequent stages identifies landmarks. The current state-of-the-art model is based on Fully Convolutional Networks[3], with PCKh (87%) on the MPII dataset [1]. We use pretrained *MMPose* [5] toolbox library which implements HRNet full body keypoints detection model [9] and *Blazepose*[2], which is a lightweight implementation of HourGlass style network architecture to generate position embeddings in our pipeline.

3 UNPOSE

The Unpose system presents an unsupervised method for classifying, and ranking most popular poses, and further extends the framework to detect the missing images from target products. The system attempts to:

(1) Define the ideal set of images in a reference set.
(2) Detect the missing images from the subject imageset *w.r.t.* to the reference set.
(3) Rank the identified images in decreasing order of importance.

The test product *subject* imageset mentioned here has less than p-threshold(p=5) number of images in the listing and is our target for improvement. We discuss this scheme's components in following sections starting by building the reference imageset.

3.1 Building reference imageset

We built the reference imageset by selecting top-K product listings by popularity belonging to *Clothing* sorted by number of customer average review ratings and number of reviews. This sorting criteria is easily configurable based on available signals.

3.2 Reference set processing flow

The reference set images selected like described in the previous section are then passed through a data pipeline (Figure 3) which

(1) Reads the image.
(2) Estimates the coordinates of various human body landmarks like *nose, shoulders, knees, wrists, elbows* etc. w.r.t. the image.
(3) Normalizes each of these coordinates w.r.t. to image size.
(4) Creates a vector representing one image from the imageset using the normalized coordinates.
(5) Creates clusters of the vector representation of all the images using K-Means algorithm to detect K centroids, where each

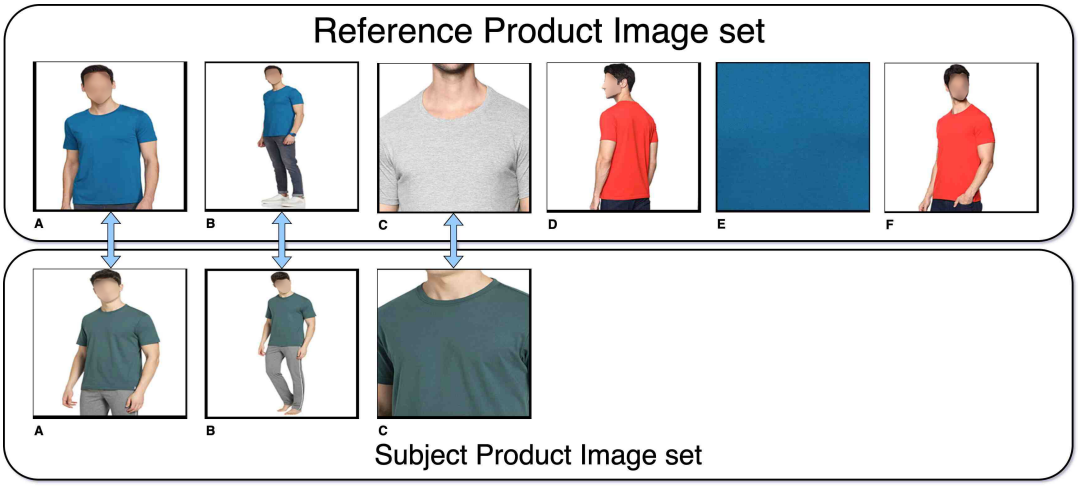

Figure 2: Reference vs Subject product image set comparison | §1.1

centroid corresponds to cluster of similar poses detected in all the images in the train imageset.

(6) Ranks all the centroids for each cohort grouped by attributes like category, subcategory, product type based on occurrence in most popular / best rated products to determine importance of each centroid within the cohort.

3.3 Detecting the missing images in Test Set

For a given product, all the images are collected as a subject product imageset. To detect the missing images from this subject imageset w.r.t. reference imageset:

(1) The model first generates the landmarks / pose embeddings coordinates for all images using the same estimation method as reference dataset.
(2) Then after normalizing these coordinates w.r.t. to image size, the vector embeddings are created.
(3) For each image, the model then finds the nearest centroid corresponding to approximate closest pose w.r.t. reference poses detected in the previous section.
(4) Further each centroid for which there were no images in the subject imageset, is ranked as per popularity criteria defined while training the reference set. Then these missing poses are flagged in the decreasing order of popularity or usefulness as illustrated in Figure 6.

3.4 Proposed image candidates

After finding the indices / labels corresponding to missing centroid in the inference flow mentioned in previous section, the user can refer the reference imageset to find out the reference images corresponding to these centroids for a given Product.

3.5 Formulation

Given a set $\mathbf{Q_D} = \{q_1^D, q_2^D ..., q_n^D\}$ of n images for a subject product $\mathbf{D}$ in imageset and a universal set $\mathbf{U} = \{u_1, u_2, ..u_m\}$ of all the m images of Top-K products in the training product imagesets, the problem is to determine a set $\mathbf{T}$ of tags, $1 \leq i \leq o$, corresponding to $\mathbf{o}$ optimal clusters $C_1, C_2, ..., C_o$ for images from $\mathbf{U}$ and further find a set $t_j \subseteq \mathbf{T}$, $1 \leq j \leq o$ of tags, corresponding to images in subject imageset $q_{t_j}^D \mid q_{t_j}^D : q_{t_j}^D \notin \mathbf{Q_D}, 1 \leq j \leq o$.

To calculate these we first determine the ideal $\mathbf{o}$-clusters from the images in $\mathbf{U}$ representing all images of all Product in reference training sets. Let $\{x_1, x_2, ..., x_G\} \in \mathbf{X_v}$ be the vector representation of any image in $\mathbf{U}$, then we determine μ_o vectors representations of these $\mathbf{o}$-clusters such that we minimize:

$$\sum_{k=0}^{m} \sum_{i=0}^{G} \min_{j \in \{0,o\}} (\| x_i - \mu_j \|_m^2)$$

Here the vector $\mathbf{X_v}$ is created by using normalized position coordinates of Point of Interests (PoI) from Pose Detection.

Later we determine all the tags l_j so as to maximize the probability $p(q_i \mid \mathbf{X_v})$ of image I to belong to the cluster given by tag $l_j \in \mathbf{T}$ such that

$$j = argmin(\| x_i^I - \mu_j \|_I^2) \; \forall \; x_i^I \in \mathbf{X_v^I}$$

Further we determine all tags t_j's such that, all tags that are in T but not are detected in the previous step.

$$t_j = \{t_i : t_i \in \mathbf{T}, \; t_i \notin l_i \forall \; 1 \leq i \leq o\}$$

These t_j's will be the missing poses / centroid in the subject imageset w.r.t. $\mathbf{o}$ centroids defined from reference imageset.

4 PIPELINE

The training pipeline that consists of Pose landmark detection, Embedding generation, the K-means clustering for ideal centroid detection, followed by ranking mechanism is described in the text below. We focus on the dataset creation used in the pipeline, then proceed to individual stages of processing.

4.1 Dataset

We aggregated public data from popular e-commerce website's clothing products metadata and listing images, and selected top 3000 most popular products with available image count per product > 9

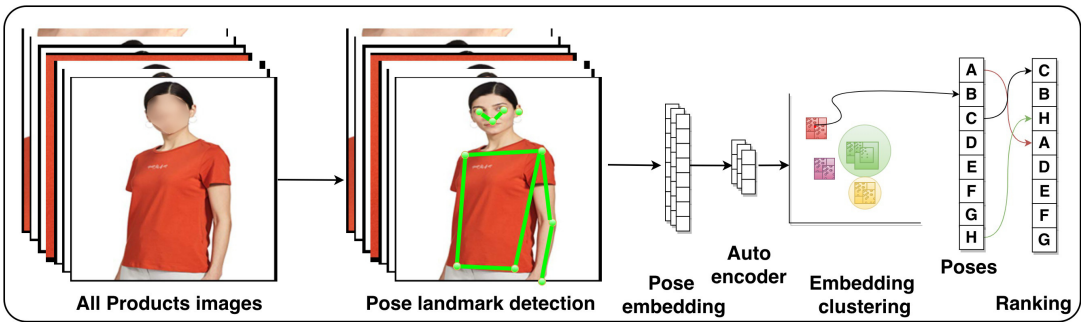

Figure 3: Training pipeline for missing image detection | §3.2

based on customer ratings, etc. This collection provided us with an imageset of ~20K for training. For evaluation purpose, we selected unseen products from the similar selection criteria collection of top products, but such that images per product imageset were < 5.

4.2 Pose Landmarks

For experimentation, the Pose Landmark coordinates are estimated using *Blazepose* pretrained implementation of HourGlass style Region detection NN library as well as *MMPose* toolkit pretrained implementation of HRNet. The implementation Blazepose [2] (MMPose [5]) generates 3D coordinates for 33 (HRNet 2D: 17) landmarks points as depicted in Figure 4. These coordinates are normalized w.r.t. images dimensions. The pose landmarks are cartesian pixel coordinates w.r.t. to image top left.

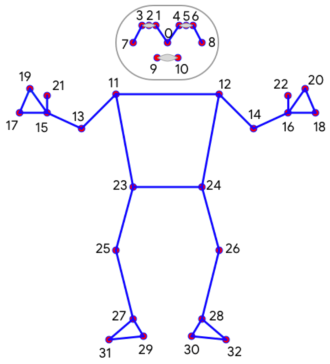

Figure 4: BlazePose landmarks[2]

4.3 Embeddings

The embedding vectors are compiled using landmark coordinates w.r.t. to image size. To add more pose related information, the embeddings are extended by taking various ratios of x, y coordinates of landmarks to detect the front, side, backview, e.g. $\frac{\mathbf{x}_{left\ shoulder}}{\mathbf{x}_{right\ shoulder}}$. Further the z components (available only for 3D-model) help identify if the image is close-up, distant pose etc. We generate a 77 (MMpose2D: 61) dimensional vector to store the features thus engineered. These embeddings are calculated for all training as well as test set images.

4.3.1 Autoencoder Embeddings. Further to develop efficient representations, the training imageset pose embeddings are fit and transformed through an *Autoencoder* to reduce the dimensionality of the embeddings, and thus learn a compressed representation of raw data to observe an improvement in clustering performance. The autoencoder optimizes for embeddings by minimizing for batchwise contrastive loss.

The Autoencoder we used is a encoder-decoder network which optimizes contrastive loss of 2048 batch size. The encoder is 4 layer fully connected feed forward network (FCFFN) with input dimensionality of embeddings from previous layer and bottleneck size of 8 nodes. The decoder is a 2 layer FCFFN. It is trained using AdamW optimizer with decay of 0.001 and starting learning rate of 0.1.

4.4 Clustering

After generating the embeddings, the pipeline further uses clustering using K-Means implementation of faiss [6] library to identify K-centroids which represents mean value of K vectors over the training image set embeddings. The centroid and clusters thus generated signify the embeddings corresponding to the most similar and recurring poses in reference image dataset. The clustering algorithm is fit over flattened representations of all images in the training imageset.

4.5 Ranking the poses

After calculating the centroids for the clusters, we determine the suitability order of each of the centroids, which in turn correspond to the poses. To quantify this suitability, for each image in the reference dataset, we estimate the closest pose (represented by the index value of cluster centroid, a value $[0, Num_{poses})$), and then rank these centroids against the target variable which is weighted mean of image's Product's customer average review rating, number of ratings etc. To account for the effect of category, subcategory, and other categorical attributes, we train a GBDT based regressor with these features along with centroids to predict the target variable. This gives us a relative order of a centroid for a given category, subcategory, etc. The Algorithm 1 summarizes the scheme.

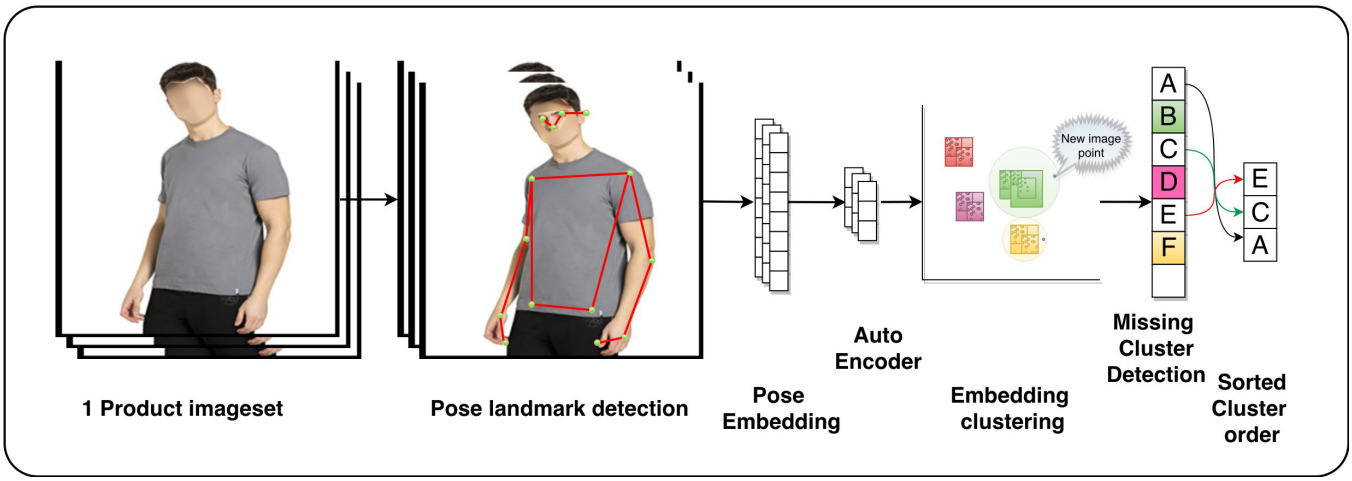

Figure 5: Inference pipeline for missing image detection | §4.6

4.6 Inference

The inference pipeline (Figure 5) follows the same process of PoI detection, embedding generation. However, inference is calculated at per Product listing - imageset. The pipeline calculates the index of nearest centroid corresponding to each image in the imageset. The inference flow is outlined in the Algorithm 2.

Algorithm 1 Training flow | §4.6

AllCentroids ← {}
AllImageVector ← [ ]
RankVector ← None
for i = 0 to len(AllTrainingImages) **do**
  landmarks ← GetLandmark(AllTrainingImages[i])
  embedding ← GetEmbedding(landmarks)
  embedding, AutoencVector ← Autoenc(embedding)
  AllImageVector[i] ← embedding
end for
AllImageVector, AutoencVector ← Autoenc(AllImageVector)
AllCentroids ← CalcAllCentroids(AllImageVector)
RankVector ← LearnRank(AllCentroids, AllTrainingImages-Data)
  return AllCentroids, RankVector, AutoencVector

The MissingCentroids thus obtained gives the indices of the Centroids corresponding to the pose w.r.t. AllCentroids representing the reference set centroids and are determined during the training phase. In Figure 6 we observe that $\{D, E, F\}$ are the detected as the missing Centroids in decreasing order of importance, and user can refer the corresponding images from the reference set.

5 PERFORMANCE AND EVALUATION

The performance of the model should be evaluated on at least 2 parameters:

(1) **Quality of clusters generated** - Are the clusters generated for true images cover all variations of images as per cataloguing guidelines.

Algorithm 2 Inference flow | §4.6

haveCentroids ← {}
for i = 0 to len(ImageSet) **do**
  landmarks ← GetLandmark(ImageSet[i])
  embedding ← GetEmbedding(landmarks)
  embedding ← Autoenc(AutoencVector, embedding)
  centroid ← GetNearestCentroid(embedding)
  haveCentroids ← haveCentroids || centroid
end for
MissingCentroids ← {}
for i = 0 to len(AllCentroids) **do**
  if i is not in haveCentroids **then**
    MissingCentroids ← MissingCentroids || i
  end if
end for
MissingCentroids ← SortCentroidsByRank(RankVector, MissingCentroids, ProductMetatData)
  return MissingCentroids

(2) **Accuracy of clustering of the validation set** - As compared to human labelled images, is clustering enumerating the missing images correctly or not. This in turn depends on the definition of embedding quality.

Item 1 is not quantitatively assessed in the text, however few examples of cluster labels are presented for subjective evaluation in Table 2. This can be addressed quantitatively in future work based on aesthetic evaluations. Item 2 is assessed based on manual labelling the validation set images. This is achieved by comparing imageset cluster labels predicted by the model vs. manually labelling an image w.r.t. cluster label of most similar image with similar pose from training set images.

where the Accuracy is defined as:

$$Accuracy = \frac{Total\ num.\ of\ Labels\ detected\ missing\ by\ model}{True\ num.\ of\ Labels\ missing\ as\ per\ human\ label}$$

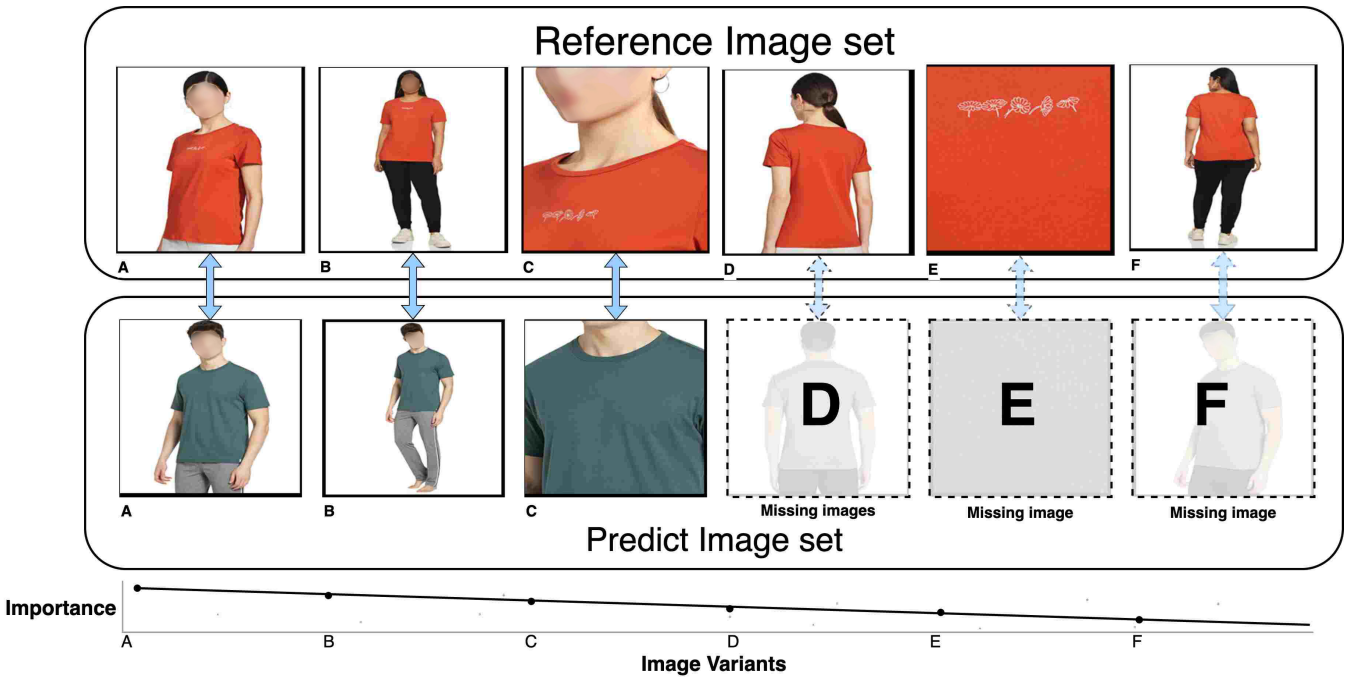

Figure 6: Sample imagesets | §3.3, §4.6
The figure shows how the ideal *Reference* imageset has centroids A-F and in the *Predict* imageset only centroids A-C are present, with MissingCentroids indicating $\{D, E, F\}$ to be the missing images in decreasing order of importance. Please refer Appendix appendix A for more examples.

Table 1: Variant label model accuracy on different subcategory under Clothing category | §5

	BlazePose		MMPose
Subcategory	**Accuracy**	**Accuracy [Autoenc]**	**Accuracy [Autoenc]**
Polo shirts	0.6967	0.8264	**0.8536**
T-shirts	0.9186	**0.9652**	0.9565
Men casual shirts	0.8442	0.8823	**0.91089**

The Table 1 shows accuracy numbers for PoI coordinates based embedding *Accuracy* and PoI coordinates along with Autoencoder reduced embeddings *[Accuracy[Autoenc]]* for BlazePose implementation of the flow. The column MMPose highlights the Accuracy numbers when embeddings were generated using MMPose2D pose estimation PoI coordinate values. The high accuracy numbers may be misleading because validation set imagesets are having very few images / repeated images, hence large num of centroids are recommended as missing, however this imageset-wise accuracy does count for correct label assignment to each individual image and is certainly indicative of effectiveness of the approach. Each subcategory set in the Table 1 had 20 product image sets with 4 or fewer images.

6 MODEL IMPLEMENTATION NOTES

We observed a few limitations in the current work with the implementation which may or may not be influence business use case, they are as follows:

- Blazepose model is not able to output non-trivial coordinates when either of limbs or head is not visible, but MMPose is able to determine coordinates if some boundary structure is available. E.g. in a t-shirt image if neck and above region cropped MMpose is able to identify it as pose. Refer Figure 9 in Appendix.
- Both Blazepose and MMPose models are not able to distinguish between male and female posers. This is generally not a con as it is seldom required to have both male/females poses in the image set.
- Behavior of model is unpredictable when there are more than one posers/apparels in the image.
- These models classify all non-human image under 1 centroid. E.g. size-chart, fabric view etc. will be grouped as same pose.

7 CONCLUSION

The text discusses a novel and promising approach to recommend images to populate image in listings for a high volume and popular category. This model can be one of the many approaches required to address different aspects like catalogue images with *pose, lighting, color, crop* etc. issue in the image listings. To improve the coverage and accuracy of the model, the processing and data set generation can be improved by adequate data exploratory analysis. The

mechanism suggested is free of human bias in the dataset if any, and recommendations thus generated are based on statistics of end user preferences, and product offering by the seller and is easily customizeable for any e-commerce website, given available levels of telemetry data.

REFERENCES

[1] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2014. 2D Human Pose Estimation: New Benchmark and State of the Art Analysis. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*. arXiv, usa, 3686–3693. https://doi.org/10.1109/CVPR.2014.471

[2] Valentin Bazarevsky, Ivan Grishchenko, Karthik Raveendran, Tyler Zhu, Fan Zhang, and Matthias Grundmann. 2020. BlazePose: On-device Real-time Body Pose tracking. https://doi.org/10.48550/ARXIV.2006.10204

[3] Adrian Bulat, Jean Kossaifi, Georgios Tzimiropoulos, and Maja Pantic. 2020. Toward fast and accurate human pose estimation via soft-gated skip connections. https://doi.org/10.48550/ARXIV.2002.11098

[4] Abon Chaudhuri, Paolo Messina, Samrat Kokkula, Aditya Subramanian, Abhinandan Krishnan, Shreyansh Gandhi, Alessandro Magnani, and Venkatesh Kandaswamy. 2018. A Smart System for Selection of Optimal Product Images in E-Commerce. https://doi.org/10.48550/ARXIV.1811.07996

[5] MMPose Contributors. 2020. OpenMMLab Pose Estimation Toolbox and Benchmark. https://github.com/open-mmlab/mmpose.

[6] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with GPUs. https://doi.org/10.48550/ARXIV.1702.08734

[7] Wenbo Li, Zhicheng Wang, Binyi Yin, Qixiang Peng, Yuming Du, Tianzi Xiao, Gang Yu, Hongtao Lu, Yichen Wei, and Jian Sun. 2019. Rethinking on Multi-Stage Networks for Human Pose Estimation. https://doi.org/10.48550/ARXIV.1901.00148

[8] Alejandro Newell, Kaiyu Yang, and Jia Deng. 2016. Stacked Hourglass Networks for Human Pose Estimation. https://doi.org/10.48550/ARXIV.1603.06937

[9] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. 2019. Deep High-Resolution Representation Learning for Human Pose Estimation. https://doi.org/10.48550/ARXIV.1902.09212

[10] Wei Tang, Pei Yu, and Ying Wu. 2018. Deeply Learned Compositional Models for Human Pose Estimation. In *Computer Vision – ECCV 2018 - 15th European Conference, 2018, Proceedings (Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics))*, Vittorio Ferrari, Cristian Sminchisescu, Martial Hebert, and Yair Weiss (Eds.). Springer Verlag, USA, 197–214. https://doi.org/10.1007/978-3-030-01219-9_12 Funding Information: Acknowledgement. This work was supported in part by National Science Foundation grant IIS-1217302, IIS-1619078, and the Army Research Office ARO W911NF-16-1-0138. Funding Information: This work was supported in part by National Science Foundation grant IIS-1217302, IIS-1619078, and the Army Research OfficeAROW911NF-16-1-0138. Publisher Copyright: © 2018, Springer Nature Switzerland AG.; 15th European Conference on Computer Vision, ECCV 2018 ; Conference date: 08-09-2018 Through 14-09-2018.

A SAMPLE OUTPUTS

This section shows some experiment results samples.

Since Unpose uses unsupervised clustering we provide an approximate human description of the images corresponding to centroids detected using MMPose2D flow in the Table 2 below. This is curated by manually inspecting many batches of images labelled by the model.

Table 2: Approximate text description of cluster obtained

Centroid Label	Approx. Human Description	Image
0	Upper body till waist | nose visible | front	
1	Upper body till waist | half | back	
2	Full body | knee bent | front	
3	Close up | chin visible | chest	
4	Full body	
5	Non-human | tables | fabric closeup	
6	Chin to torso	
7	Full body	

In the sections below we look at different subcategory of apparel and labels identified for prexisting images in the Product imageset and missing image labels proposed by the model.

A.1 Polo T-Shirt

Figure 7:
The figure shows the existing imageset for an Product and the labels predicted by the model. The numbers below the images correspond to the centroid index numbers from the Table 2. The missing labels proposed by Unposed were: 4, 6, 7 | Refer Table 2

A.2 T-Shirt

Figure 8:
The figure shows the existing imageset for an Product and the labels predicted by the model. The numbers below the images correspond to the centroid index numbers from the Table 2. The missing labels proposed by Unposed were: 0, 2, 3, 4, 7 | Refer Table 2

Figure 9:
The figure shows the labels predicted by the MMPose model, the numbers under the images correspond to the centroid index numbers predicted for the existing imageset.
The model predicts classes 0 and 1 corresponding to front and back poses as per Table 2, though human keypoints are missing. This may acceptable for the use case as the desired poses are depicted.
The resultant missing labels proposed by Unposed were: 2, 3, 4, 7 | Refer Table 2 | §6