

# Content-based product image retrieval using squared-hinge loss trained convolutional neural networks

Arif Rahman<sup>1,2</sup>, Edi Winarko<sup>1</sup>, Khabib Mustofa<sup>1</sup>

<sup>1</sup>Department of Computer Science and Electronics, Universitas Gadjah Mada, Yogyakarta, Indonesia

<sup>2</sup>Department of Information System, Universitas Ahmad Dahlan, Yogyakarta, Indonesia

---

## Article Info

### Article history:

Received Sep 18, 2022

Revised Feb 13, 2023

Accepted Mar 9, 2023

---

### Keywords:

Content-based  
Convolutional neural network  
Image retrieval  
Product image  
Squared-hinge loss

---

## ABSTRACT

Convolutional neural networks (CNN) have proven to be highly effective in large-scale object detection and image classification, as well as in serving as feature extractors for content-based image retrieval. While CNN models are typically trained with category label supervision and softmax loss for product image retrieval, we propose a different approach for feature extraction using the squared-hinge loss, an alternative multiclass classification loss function. First, transfer learning is performed on a pre-trained model, followed by fine-tuning the model. Then, image features are extracted based on the fine-tuned model and indexed using the nearest-neighbor indexing technique. Experiments are conducted on VGG19, InceptionV3, MobileNetV2, and ResNet18 CNN models. The model training results indicate that training the models with squared-hinge loss reduces the loss values in each epoch and reaches stability in less epoch than softmax loss. Retrieval results show that using features from squared-hinge trained models improves the retrieval accuracy by up to 3.7% compared to features from softmax-trained models. Moreover, the squared-hinge trained MobileNetV2 features outperformed others, while the ResNet18 feature gives the advantage of having the lowest dimensionality with competitive accuracy.

*This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.*



---

## Corresponding Author:

Edi Winarko

Department of Computer Science and Electronics, Universitas Gadjah Mada  
Building C, 4<sup>th</sup> Floor, Sekip Utara, Bulaksumur, Yogyakarta 55281, Indonesia  
Email: ewinarko@ugm.ac.id

---

## 1. INTRODUCTION

Users input a text containing the product's keywords to search for an item in an online store. However, text keywords could not distinguish products based on their visual perception. A visual search system uses an image as a query instead of a text to solve the problem. Visual features representing the image are extracted and matched with others in the database to get similar images [1]. Image features represent shapes or color distribution of the image. Shape-based image retrieval uses edges or moment invariants, while color-based retrieval uses a histogram of the image pixel values. Key-point-based features, including scale-invariant feature transform (SIFT) [2] and speeded-up robust features (SURF) [3], are also used in visual searches besides those features.

Recent studies have focused on convolutional neural networks (CNN) since it outperforms other approaches in large-scale object detection and image classification [4] in ImageNet large scale visual recognition challenge (ILSVRC) competitions [5]. Furthermore, the CNN model is also applied in image matching and retrieval. CNN-based features of the image were obtained by training the CNN model for the image classification task using a specific dataset and a loss function. Then, the model is modified by removing its classification layer. After that, an input image is processed with the modified model to extract the feature [6].

CNN-based features have been applied in content-based product image retrieval. The CNN model is trained for classification tasks using product category label supervision. The output is the model's last layer trained using the softmax loss function for multiclass classification. Along with softmax loss, the squared-hinge loss function is also known for its performance in multiclass classification [7]. Unfortunately, to our knowledge, studies on content-based retrieval, specifically in product images, have not applied the squared-hinge loss function in their model training.

This study proposes a method for extracting features from product images based on CNN models trained with squared-hinge loss as an alternative to softmax loss. Extracted image feature was then indexed using the nearest-neighbour (NN) indexing technique. Retrieval experiments were conducted on different CNN models with softmax and squared-hinge loss functions. We evaluate the training process and the retrieval result to obtain the best configuration for the feature extraction method.

Contributions of this study are: i) Our method can be used as an alternative to the existing CNN-based feature extraction method, specifically for content-based product image retrieval. Also, ii) we present the best configuration of the CNN model, training parameter, and loss function to achieve the best result, and iii) We believe that our works can be applied for content-based retrieval in e-commerce shops.

The rest of the paper is organized as follows. Section 2 reviews related works on CNN features and product retrieval. Details on the method for feature extraction, indexing, and matching are given in section 3. Experimental results with discussion are presented in section 4. Finally, in section 5, we conclude the paper with a summary.

## 2. CNN-BASED FEATURES

A deep convolutional network consists of two parts: i) the convolution layer and the pooling layer and ii) the fully connected layer [8], as shown in Figure 1. The first layer in the convolutional layer is the input layer that accepts input in a raw red, green, blue (RGB) pixel image. A convolutional layer is a set of feature maps with neurons, and the parameters of the layer are the filter or kernel set. The pooling layer reduces the activation map's spatial dimension, the number of parameters in the network, and computational complexity. The fully-connected layer (FC) has neurons connected to the previous layer as in a neural network [9]. A loss value is a penalty for a mismatch between the desired output and the resulting output of the last layer. CNN feature extracted from the nodes in the FC since this layer has a global receptive field representing the global features [10].

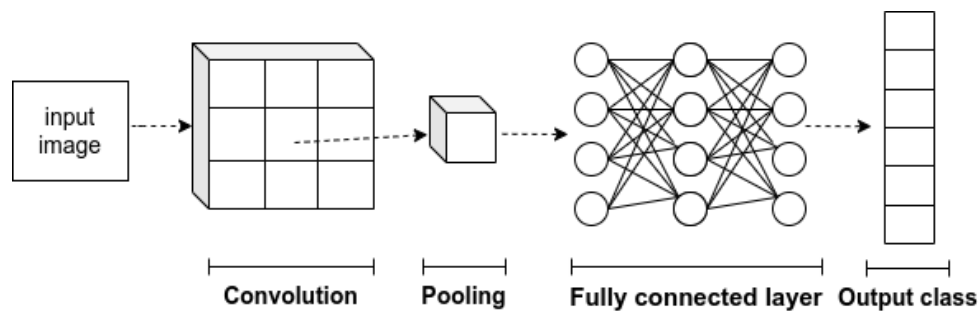


Figure 1. Convolutional neural network architecture

Researchers use various CNN models for feature extraction. For example, in [11], [12], use the FC before the last layer, named FC6 and FC7 of the AlexNet model [13]. While Razavian *et al.* [14] applies a different CNN model, i.e., OverFeat [15], and extract feature from the first FC layer (layer 22) of the model. Moreover, the HybridNet [16] was used in [17] to extract features using the activation of the first FC layer (i.e., FC6).

CNN feature extraction described above is performed on general image classification and retrieval. Researchers were also interested in applying the CNN feature extraction method to specific images, such as product images. The work of [18] uses a self-built network model, which is more straightforward than the standard pre-defined model for classifying images. In [19], product retrieval using CNN features from the VGG-19 model was applied to retrieve fashion product images. Alternatively, Elleuch *et al.* [20] used the features from the Inception V3 model [21]. The feature is extracted from the bottlenecks layer, the layer just before the last output layer on a clothing dataset.

### 3. METHOD

The method in this study consists of three steps, as shown in Figure 2. The transfer learning scheme is applied to the pre-trained CNN model in the model training. Then, the model was trained with images and category labels from the dataset using squared-hinge loss. The output of this step is the fine-tuned CNN model. After that, image features are extracted from the fine-tuned CNN model and indexed using the nearest neighbor (NN) indexing technique. In image retrieval, the query-by-example is performed by matching features from the query with the database's indexed features using the k-NN search algorithm.

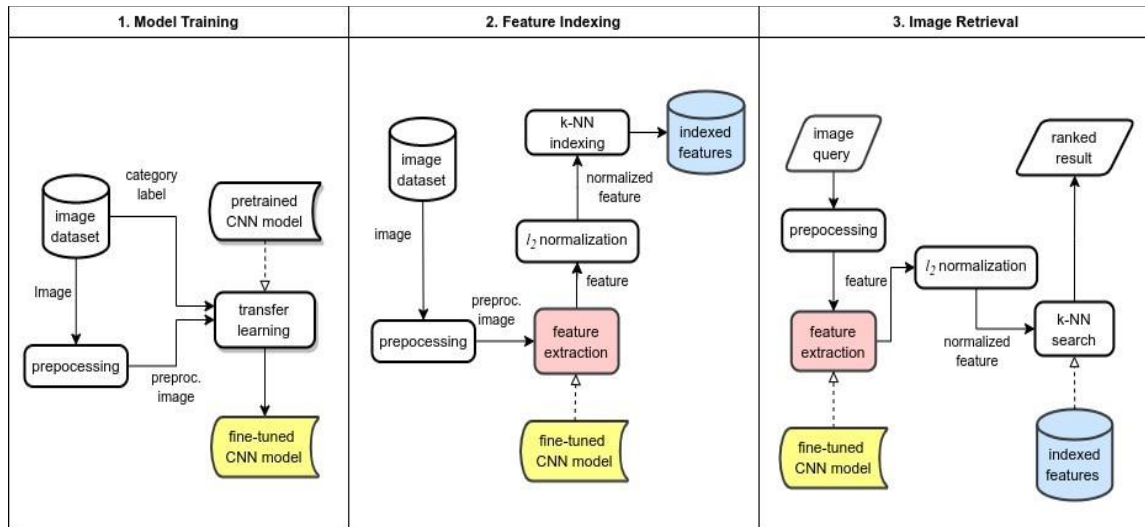


Figure 2. The proposed method for content-based product image retrieval

#### 3.1. CNN models

The CNN architecture has evolved, particularly in the convolutional module. The VGG19 [21] is one model that uses the conventional convolutional module. Another model uses several filter sizes in a single image block, then concatenated and transferred onto the next layer instead of limited to a single filter size, as in the Inception module [22]. MobileNetV2 [23] uses depthwise convolution to reduce the number of parameters, which applies a single convolutional filter per input channel. A pointwise convolution is used to create a linear combination of the output of the depthwise convolution. Unlike the types above, ResNet [24] uses the residual module, a skip-connection block that learns residual functions with reference to the layer inputs instead of learning unreferenced functions.

The layer before the last fully connected layer ( $FC_{n-1}$ ) represents the image feature vector extracted from the CNN model. Each node in the layer reflects a feature vector element. Therefore, the number of nodes in the  $FC_{n-1}$  should be considered when selecting the CNN model for the image feature extractor since it affects the feature dimension. We use CNN models with different  $FC_{n-1}$  nodes, as shown in Table 1, to see the correlation between the feature dimension and retrieval accuracy.

Table 1. Main module and  $FC_{n-1}$  nodes of CNN models

No	Model	Main module	$FC_{n-1}$ nodes
1	VGG-19	convolution	4,096
2	InceptionV3	inception	2,048
3	MobileNetV2	depthwise convolution	1,280
4	ResNet-18	residual	512

#### 3.2. Transfer learning on the CNN model

Deep transfer learning tries to improve accuracy by adopting the model from another domain that has high accuracy. We perform transfer learning from a pre-trained ImageNet model [5] as the source domain to the product images dataset as the target domain. First, the source model's convolution and pooling layer parameters were transferred to the target as in the network-based transfer learning schema [25]. Then, to fine-tune the model, the source model's last FC layer ( $FC_n$ ) was replaced by a new layer ( $FC'_n$ ). This new layer has the same number

of nodes as the number of classes in the product dataset. After that, the fine-tuned model is retrained using the product image dataset. Figure 3 shows the transfer learning process from the pretrained model.

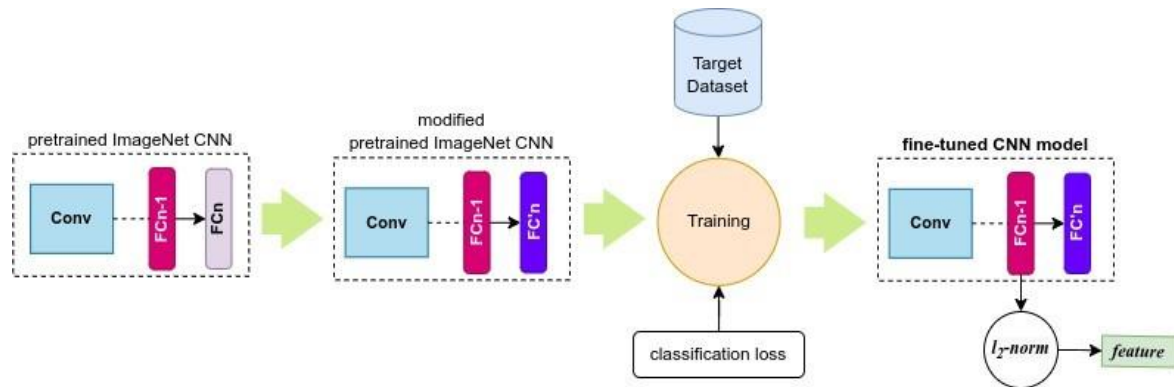


Figure 3. Transfer learning process of CNN model

### 3.3. Image pre-processing

Before an image is processed in training and retrieval, pixel normalization is needed since the image has a considerable variation of pixel values. Normalization ensures the values are within a specific range and reduces skewness, which helps the model learn faster and better. The pixel value is normalized using the  $z$ -score in (1).

$$z'_i = \frac{z_i - \mu}{\sigma} z_i \tag{1}$$

where  $z'_i$  is the normalized value, and  $z_i$  is the pixel value on channel  $i$  (i.e., R, G, and B). The  $\mu$  and  $\sigma$  are the mean and standard deviation of the pixel values, respectively. Figures 4(a) and 4(b) show examples of the images before and after normalization taken from Stanford online product (SOP) and InShop DeepFashion (InShop).



Figure 4. Examples of original (left) and normalized (right) images in (a) SOP [26] and (b) InShop [27] datasets

We also perform image data augmentation, a technique to supplement the images dataset for training. This study used two types of image augmentation, i.e., fixed-sized cropping on random locations and random horizontal flipping. Those augmentations are applied so the model can learn images in various positions during training. Hence the model is more robust against image position variation.

### 3.4. Loss function

We trained the CNN model using a loss function for classification tasks. Since it is a multiclass problem, the loss function should produce multiclass outputs. Softmax loss  $L_S$  in (2) is typical for a multiclass classification task.

$$L_S = -\frac{1}{N} \sum_{i=1}^n \log \left( \frac{e^{f_{yi}}}{\sum_{j=1}^K e^{f_j}} \right) \quad (2)$$

$f_{yi}$  is output from fully-connected layer  $f$  for input with label  $y_i$ , while  $f_j$  for label  $j$ ,  $N$  is the number of samples, and  $K$  is a total class number.

Alternatively, for multiclass classification, we can use squared-hinge loss  $L_H$  expressed in (3), where  $m$  is the specified margin value.

$$L_H = \frac{1}{N} \sum_{i=1}^n \sum_{i \neq j}^K \max(0, m - f_{yi} + f_j) \quad (3)$$

The  $L_H$  is a more local objective since it computes uncalibrated scores for all classes, while  $L_S$  allows all labels' computing probabilities.

### 3.5. Feature extraction and indexing

Feature extraction is the inference process of the fine-tuned CNN model. The feature extracted is a CNN-based feature, a vector whose value is the node value of the fully-connected layer before the last layer in the model  $FC_{n-1}$ . This layer has a global receptive field that can be used as a global image feature [10]. The vector feature was then normalized with  $l_2$ -norm as in (4) to form the image feature  $F$ . This normalization ensures that vector feature values are in the specific range of value.

$$F = \frac{FC_{n-1}}{\|FC_{n-1}\|_2} \quad (4)$$

The vector feature values are stored in a feature database before being used in the image-searching process. An indexing technique was applied to store and retrieve features efficiently. The nearest-neighbor indexing is applied since it is straightforward and sufficient for image search with CNN-based features.

### 3.6. Image retrieval

The retrieval process begins with preprocessing the image query. Then, the image feature is extracted using the fine-tuned CNN model from the model training. After that, the extracted feature is normalized using  $l_2$  normalization as in (4). This normalized feature is used as the query to get similar image features in the indexed feature database. The feature similarity is measured based on the Euclidean distance of the query and each image feature in the database. Similarity search of the feature on the database performed using the k-NN search algorithm. The  $k$  feature vectors in the indexed feature database with the smallest distance to the query are returned as the query result.

## 4. RESULTS AND DISCUSSION

### 4.1. Experimental setup

We use hierarchical class image data with general to specific levels: superclass, class, and image. A superclass refers to product categories (e.g., bicycle, sofa, and shirt), the class represents each product item in the product category, and an image is the product item picture. A product item may contain more than one picture. The experiments were conducted using labeled product image datasets with fine-grained categories: Stanford online product (SOP) [26] and InShop DeepFashion (InShop) [27]. The SOP includes home product images, while InShop contains clothing images. SOP consists of 12 superclasses, 22,634 classes, and 120,053 images. The InShop dataset contains 23 superclasses, 7,982 classes, and 52,712 images. Both SOP and inShop datasets are also used in [28].

CNN model training was performed on a GPU-enabled machine in 100 epochs. The training is optimized with stochastic gradient descent (SGD), and the learning rate is set to  $10^{-3}$ . Feature indexing and k-NN searching are implemented using neighborhood graph and tree (NGT) [29], a graph-based indexing library. Results on diverse datasets in [30] show that the batch size at the training CNN model gets the best accuracy at sizes greater than 64. However, the greater batch size requires more computational resources. Due to the limitation of computational resources, we found that 96 is the optimum batch size in our experiments.

## 4.2 Evaluation metrics

Retrieval results on both datasets using queries from all images in the test split are evaluated with two metrics, and for both, a higher value means better performance.

- The  $P @ k$  (precision at  $k$ ) is expressed as (5),

$$P@k = \frac{\sum_{q \in Q} \text{hit}(q,k)}{|Q|} \quad (5)$$

- The  $mAP @ k$  (mean average precision at  $k$ ) is written as (6),

$$mAP@k = \frac{1}{|Q|} \sum_{q \in Q} \frac{1}{r_i} \sum_{i=1}^k \text{hit}(q, i) \quad (6)$$

where  $r_i$  is the number of relevant results in the top  $i$ . This metric considers the order of the retrieval results. We use  $k=50$  in the experiments, assuming maximum query results displayed to the user when searching for an online shop item.

## 4.3. Model training

The CNN model training was conducted using the squared hinge and softmax loss functions. We use a learning graph with loss value and epoch axes to analyze training performance. In general, the training obtains good-fit models for all datasets since the curves in the graphs descend smoothly and converge to a certain point. Figure 5(a) shows that the models trained with squared-hinge loss in the SOP dataset tend to have lower graph curves than softmax. Also, the loss values of the squared hinge descended to the point of stability before the softmax. The training in the InShop dataset gets a similar result, as shown in Figure 5(b). However, the gap between the two loss graphs is more considerable. The experiment results indicate that training the model using square hinge loss has advantages since it reduces the loss values in each epoch and reaches stability in less epoch than softmax loss.

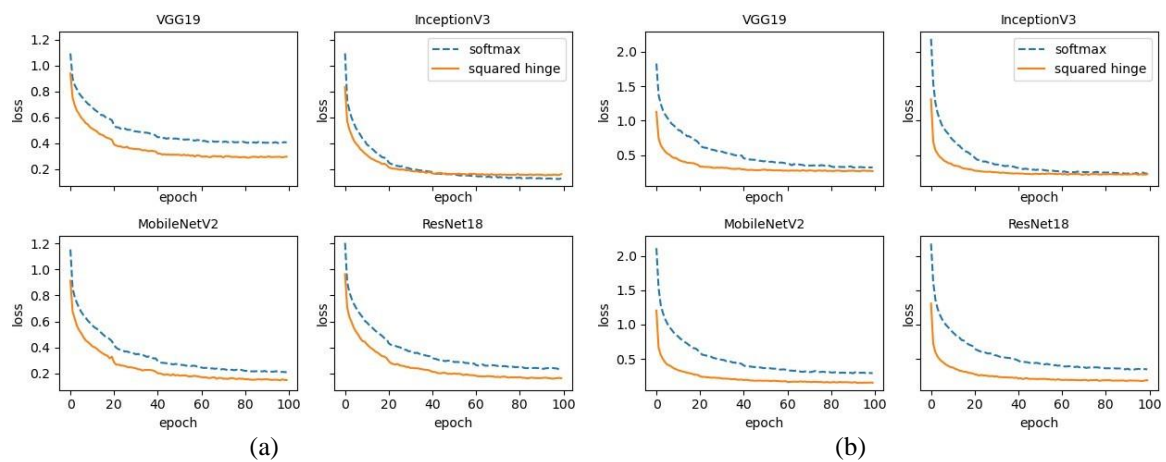


Figure 5. Loss graph of training CNN models in (a) SOP and (b) InShop dataset

The time consumed for training the models was measured in experiments. Table 2 shows the average time required for training the model per epoch. These results show that training with squared hinge loss consumed slightly more time than softmax on average at 18.75 and 65.67 seconds in InShop and SOP datasets, respectively.

Table 2. CNN models average training time per epoch (in seconds)

No	Model	InShop		SOP	
		Softmax	Squared-hinge	Softmax	Squared-hinge
1	VGG-19	477	483	1,063	947
2	InceptionV3	172	199	389	554
3	MobileNetV	104	135	307	312
4	ResNet-18	65	76	148	175

#### 4.4. Retrieval results

Retrieval experiments were performed on CNN-based features using SOP and InShop datasets using CNN models trained with softmax (S) and squared-hinge (H) loss. In Figure 6(a), P@k results on retrieval using features from ResNet18-H and MobileNetV2-H tend to get better accuracy in the SOP dataset. Besides, in Figure 6(b), retrieval results in mAP@k metric on ResNet18-H and MobileNetV2-H have very slightly different values, and ResNet18-H gets the best results in each  $k$ .

The findings for the InShop dataset retrieval are comparable to those of the SOP dataset. Figure 7(a) and Figure 7(b) show the P@k and mAP@k results on the InShop dataset. ResNet18-H and MobileNetV2-H continue to exhibit the highest accuracy in P@k and mAP@k. However, the gap between the two is now more pronounced, with MobileNetV2-H delivering superior performance compared to the others.

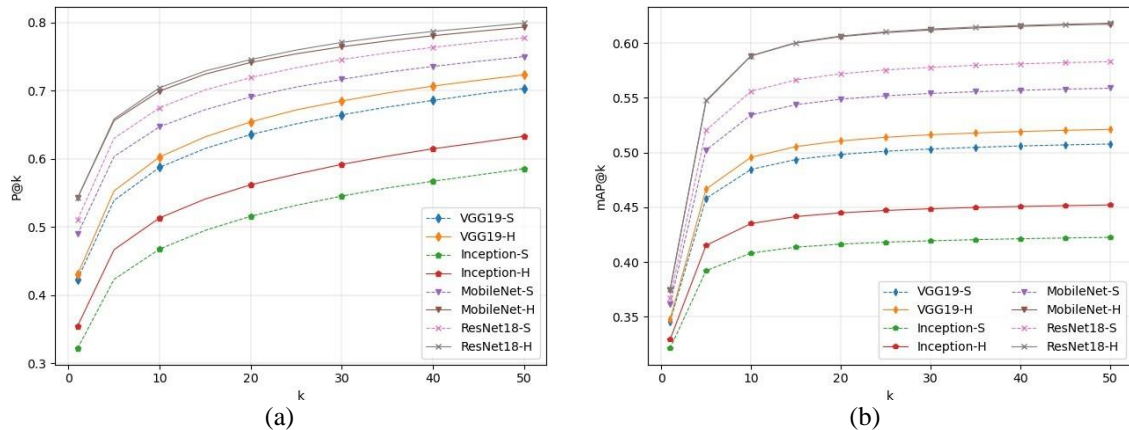


Figure 6. Retrieval results of CNN-based features trained with softmax loss (S) and squared-hinge loss (H) on the SOP dataset in (a) P@k and (b) mAP@k metric

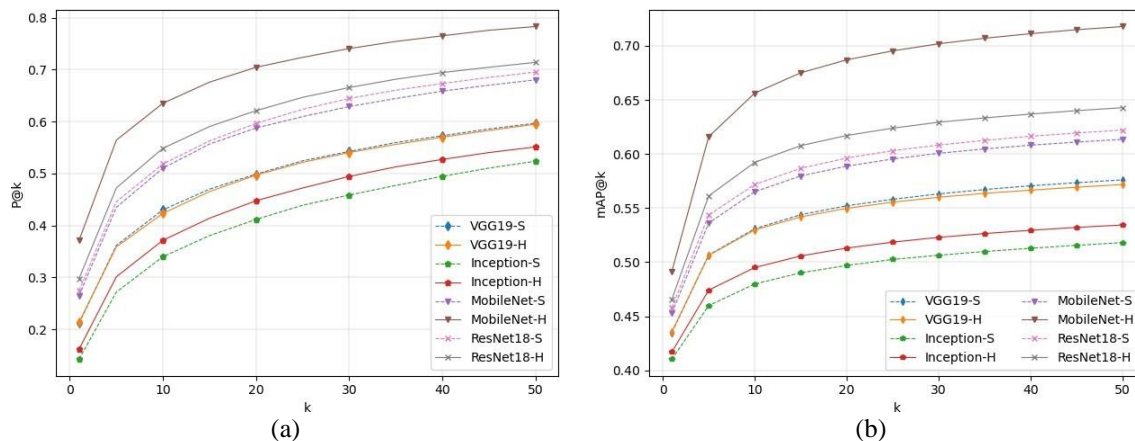


Figure 7. Retrieval results of CNN-based features trained with softmax loss (S) and squared-hinge loss (H) on InShop dataset in (a) P@k and (b) mAP@k metric

We calculated the P@k and mAP@k accuracy gap between squared-hinge loss, and softmax loss trained CNN feature. In all models and  $k=1$  to 50, retrieval accuracies using squared-hinge loss trained feature improve the softmax loss trained feature by 3.3% on average in SOP, while in Inshop by 3.7%. These results confirm that utilizing the CNN feature from the model trained with squared-hinge improves accuracy compared to a model trained with softmax loss.

Feature vectors extracted from the CNN models have various dimensions, from 4096-dim in VGG19 to 512-dim in ResNet18. Feature dimension affects the computational resource requirements since extraction, indexing, and matching operations are performed on each feature vector element. From this point of view, the ResNet18 feature is preferred to other features since it has the lowest feature dimension while still giving competitive accuracy.

## 5. CONCLUSION

Image retrieval using features extracted from various fine-tuned CNN models has been done using fine-grained image product datasets. Retrieval results using our method show that features from the CNN model trained with squared-hinge loss improve the retrieval accuracy compared to features from softmax-trained models. Overall, MobileNetV2-H features get the best retrieval accuracies. However, ResNet18-H has the advantage since it has the lowest feature dimension while still getting competitive accuracy. Since the image of a product item in the online store has a different quality than the image taken by the user, distance metric learning should be considered for calculating the distance between features.

## ACKNOWLEDGMENT

This publication was supported by Rekognisi Tugas Akhir (RTA) 2022 Research Grant from the Research Directorate of Universitas Gadjah Mada. The authors also express their gratitude to EFISON Lisan Teknologi for providing HPC ALELEON Mk.II via the “EUREKA!” program.

## REFERENCES




- [1] C. Celik and H. S. Bilge, “Content based image retrieval with sparse representations and local feature descriptors: a comparative study,” *Pattern Recognition*, vol. 68, pp. 1–13, Aug. 2017, doi: 10.1016/j.patcog.2017.03.006.
- [2] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004, doi: 10.1023/B:VISI.0000029664.99615.94.
- [3] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “Speeded-up robust features (SURF),” *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, Jun. 2008, doi: 10.1016/j.cviu.2007.09.014.
- [4] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, “Deep learning for visual understanding: A review,” *Neurocomputing*, vol. 187, pp. 27–48, Apr. 2016, doi: 10.1016/j.neucom.2015.09.116.
- [5] O. Russakovsky *et al.*, “ImageNet large scale visual recognition challenge,” *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, Dec. 2015, doi: 10.1007/s11263-015-0816-y.
- [6] M. Tzelepi and A. Tefas, “Deep convolutional learning for content based image retrieval,” *Neurocomputing*, vol. 275, pp. 2467–2478, Jan. 2018, doi: 10.1016/j.neucom.2017.11.022.
- [7] Y. Tang, “Deep learning using linear support vector machines,” *arXiv:1306.0239*, Jun. 2013.
- [8] W. Rawat and Z. Wang, “Deep convolutional neural networks for image classification: a comprehensive review,” *Neural Computation*, vol. 29, no. 9, pp. 2352–2449, Sep. 2017, doi: 10.1162/neco\_a\_00990.
- [9] T. Gorach, “Deep convolutional neural networks - a review,” *International Research Journal of Engineering and Technology (IRJET)*, vol. 5, no. 7, pp. 439–452, 2018.
- [10] L. Zheng, Y. Yang, and Q. Tian, “SIFT meets CNN: a decade survey of instance retrieval,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 5, pp. 1224–1244, May 2018, doi: 10.1109/TPAMI.2017.2709749.
- [11] J. Donahue *et al.*, “Decaf: A deep convolutional activation feature for generic visual recognition,” in *ICML’14: Proceedings of the 31<sup>st</sup> International Conference on International Conference on Machine Learning*, 2014, vol. 32, pp. 1–647–1–655.
- [12] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, “Neural codes for image retrieval,” in *ECCV 2014: Computer Vision – ECCV 2014*, 2014, pp. 584–599, doi: 10.1007/978-3-319-10590-1\_38.
- [13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.
- [14] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, “CNN features off-the-shelf: an astounding baseline for recognition,” in *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Jun. 2014, pp. 512–519, doi: 10.1109/CVPRW.2014.131.
- [15] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, “OverFeat: Integrated recognition, localization and detection using convolutional networks,” *arXiv:1312.6229*, Dec. 2013.
- [16] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, “Learning deep features for scene recognition using places database,” in *NIPS’14: Proceedings of the 27<sup>th</sup> International Conference on Neural Information Processing Systems*, 2014, vol. 1, pp. 487–495.
- [17] G. Amato, F. Falchi, C. Gennaro, and F. Rabitti, “YFCC100M hybridNet fc6 deep features for content-based image retrieval,” in *Proceedings of the 2016 ACM Workshop on Multimedia COMMONS*, Oct. 2016, pp. 11–18, doi: 10.1145/2983554.2983557.
- [18] T. Liu, R. Wang, J. Chen, S. Han, and J. Yang, “Fine-grained classification of product images based on convolutional neural networks,” *Advances in Molecular Imaging*, vol. 8, no. 04, pp. 69–87, 2018, doi: 10.4236/ami.2018.84007.
- [19] L. Fengzi, S. Kant, S. Araki, S. Bangera, and S. S. Shukla, “Neural networks for fashion image classification and visual search,” *SSRN Electronic Journal*, 2020, doi: 10.2139/ssrn.3602664.
- [20] M. Elleuch, A. Mezghani, M. Khemakhem, and M. Kherallah, “Clothing classification using deep CNN architecture based on transfer learning,” in *HIS 2019: Hybrid Intelligent Systems*, 2021, pp. 240–248, doi: 10.1007/978-3-030-49336-3\_24.
- [21] C. Szegedy *et al.*, “Going deeper with convolutions,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.
- [22] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 2818–2826, doi: 10.1109/CVPR.2016.308.
- [23] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted residuals and linear bottlenecks,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp. 4510–4520, doi: 10.1109/CVPR.2018.00474.
- [24] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.
- [25] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, “A survey on deep transfer learning,” in *Artificial Neural Networks and Machine Learning*, 2018, pp. 270–279, doi: 10.1007/978-3-030-01424-7\_27.
- [26] H. O. Song, Y. Xiang, S. Jegelka, and S. Savarese, “Deep metric learning via lifted structured feature embedding,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 4004–4012, doi: 10.1109/CVPR.2016.434.
- [27] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, “DeepFashion: Powering robust clothes recognition and retrieval with rich






- annotations,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 1096–1104. doi: 10.1109/CVPR.2016.124.
- [28] A. Rahman, E. Winarko, and K. Mustofa, “Product image retrieval using category-aware Siamese convolutional neural network feature,” *Journal of King Saud University Computer and Information Sciences*, vol. 34, no. 6, pp. 2680–2687, Jun. 2022, doi: 10.1016/j.jksuci.2022.03.005.
- [29] M. Iwasaki and D. Miyazaki, “Optimization of indexing based on k-nearest neighbor graph for proximity search in high-dimensional data,” *arXiv:1810.07355*, Oct. 2018.
- [30] P. M. Radiuk, “Impact of training set batch size on the performance of convolutional neural networks for diverse datasets,” *Information Technology and Management Science*, vol. 20, no. 1, Jan. 2017, doi: 10.1515/itms-2017-0003.

## BIOGRAPHIES OF AUTHORS






**Arif Rahman**    received a Bachelor of Computer Science from Universitas Gadjah Mada, Indonesia, Master of Informatics from Institut Teknologi Bandung, Indonesia. He is a lecturer at the Department of Information Systems, Faculty of Applied Science and Technology, Universitas Ahmad Dahlan, Indonesia, pursuing his doctoral program in Computer Science at the Department of Computer Sciences and Electronics, Universitas Gadjah Mada, Indonesia. His research interest is image retrieval, machine learning, and datamining. He can be contacted at email: arif.rahman@is.uad.ac.id.



**Edi Winarko**    received his Undergraduate degree (Drs.) in Mathematics from the Faculty of Mathematics and Natural Sciences Universitas Gadjah Mada, Indonesia, an M.Sc. in Computer Science from the Queen’s University Canada, and holds Ph.D. in Computer Science from the Flinders University, Australia. He is a senior lecturer in the Department of Computer Science and Electronics, Faculty of Mathematics and Natural Sciences, Universitas Gadjah Mada, Indonesia. His current research interests are data mining, machine learning, NLP, and information retrieval. He can be contacted at email: ewinarko@ugm.ac.id.



**Khabib Mustofa**    received his Undergraduate degree (S.Si.) in Computer Science from Universitas Gadjah Mada, Indonesia, Master degree in Computer Sciences (M.Kom.) from Universitas Gadjah Mada, and Dr. techn from the Institute for Software Engineering and Interactive Systems, Technische Universit at Wien (Vienna University of Technology), Austria. He is a lecturer at the Department of Computer Science and Electronics, Faculty of Mathematics and Natural Sciences, Universitas Gadjah Mada. His research interests cover web technology (Semantic web, Web Services), mobile application, and information management. He can be contacted at email: khabib@ugm.ac.id.