Computers, Materials & Continua
DOI: 10.32604/cmc.2023.035716
Article

Tech Science Press

Multimodal Fused Deep Learning Networks for Domain Specific Image Similarity Search

Umer Waqas, Jesse Wiebe Visser, Hana Choe and Donghun Lee*

Research and Development, AItheNutrigene, Seoul, 06132, Korea
*Corresponding Author: Donghun Lee. Email: donghun.lee@aithenutrigene.com
Received: 01 September 2022; Accepted: 09 November 2022

Abstract: The exponential increase in data over the past few years, particularly in images, has led to more complex content since visual representation became the new norm. E-commerce and similar platforms maintain large image catalogues of their products. In image databases, searching and retrieving similar images is still a challenge, even though several image retrieval techniques have been proposed over the decade. Most of these techniques work well when querying general image databases. However, they often fail in domain-specific image databases, especially for datasets with low intraclass variance. This paper proposes a domain-specific image similarity search engine based on a fused deep learning network. The network is comprised of an improved object localization module, a classification module to narrow down search options and finally a feature extraction and similarity calculation module. The network features both an offline stage for indexing the dataset and an online stage for querying. The dataset used to evaluate the performance of the proposed network is a custom domain-specific dataset related to cosmetics packaging gathered from various online platforms. The proposed method addresses the intraclass variance problem with more precise object localization and the introduction of top result reranking based on object contours. Finally, quantitative and qualitative experiment results are presented, showing improved image similarity search performance.

Keywords: Image search; classification; image retrieval; deep learning

1 Introduction

In the continuously evolving industrial and technological world there has been a steady shift towards visual representations of data. Visual data such as images and videos have led to versatile online content. Especially online shopping and other similar platforms are managing large databases of product images. Visual representations of their inventory to present to customers have become substantially important over the years. Expansion of such databases comes with complexity in searchability. It can be difficult for users to find what they are looking for, but also hard to manage the database itself.

When a database of products becomes significantly extensive, users might have a hard time finding the item they are looking for. In this situation, commonly the database is categorized and labelled to improve searchability. A user can then select categories and see all items listed within, or they can search through a text-based input which returns items with categories or labels most resembling their search query. Unfortunately, it can be hard for users to guess in which category the product they are looking for might be, or it might be hard to describe what they are looking for in a short text query. It also becomes tedious for the people managing the database to label a large number of items. Furthermore, this method of search fails entirely if the items are inherently hard to categorize. The process of finding a product using these search systems on a large database takes significant time and effort.

A different approach to finding a product in such a database is an image-based search system. This is a technique of computer vision that provides a solution to search in a large image dataset based on the visual contents of a query image alone. This means that datasets of images can be searched by using only a query image, without any additional labels.

This technique has been used in applications such as "reverse image search", where a user can take a picture of an object they want to search for and retrieve the related information of the object or visually similar images from a database. Some popular implementations of these types of image-based search systems are represented in websites such as TinEye [1], Google Images [2] and Bing Image Search [3]. Although these search engines are successful implementations, they are extremely generalized. These systems are good at identifying general classes of objects; however, they are not as effective in finding very specific objects. They usually fail to retrieve satisfying results when the object a user is looking for is very sensitive to small changes in shape and design. Where an object, despite having a generally similar shape, might have a completely different purpose which can only be seen in small variations in the shape. This problem is prevalent in domain-specific datasets which have low intraclass variance and requires special attention to detail. Practical implementations for domain-specific image-based search include E-commerce product retrieval. Many large e-commerce platforms, such as Alibaba, eBay and Walmart already utilize these types of search systems. However, in these systems, the gap between expected and retrieved results still stays considerably large. Especially, in fine-grained product classes with low intraclass variance. Therefore, to address these challenges this paper focuses on a domain-specific dataset with low intra-class variance to improve query results of products in classes that are very sensitive to small changes in shape or design. The proposed approach provides a practical application of image-based search to small, medium, or large industries which maintain domain-specific datasets.

The following significant contributions are presented in this work:

- An improved object localization is proposed to better extract objects of interest with refined edges while preserving sensitive features without losing crucial object parts which play a critical role during similarity calculation. The object is extracted with state of the art models by fusing U^2-net with F, B, Alpha Matting.
- An improved classification model is proposed to narrow down the search by correctly predicting the category of the object presented in the query image to address computational cost and eliminate searching through unrelated categories.
- Reranking with the IoU metric is proposed on top of the similarity search, where the object of interest in the images are already extremely similar but might differ slightly in size or shape.
- Finally, a multimodal fusion is proposed which connects each deep learning network or module contribution to form the final image similarity search outcome. The results were compared quantitatively or qualitatively with other models or techniques used in previous image-based

search engines. Results suggest that our proposed approach performed better in domain-specific data which is extremely sensitive to change or has a low intraclass variance.

The paper is organized as follows: Section 2 presents the related work. Section 3 presents the dataset and Section 4 introduces the design and development phases of the proposed methods. The details of experiment results and performance are discussed in Section 5 and finally, Section 6 concludes the paper and recommends future direction.

2 Related Work

In datasets with fine-grained classes containing complex and noisy backgrounds, it remains challenging to find similar images to a query image. A variety of research is done on the technical implementation of image-based search. To be able to calculate the similarity of images compared to a query image it has been proven to be useful to encode the features presented in an image. Therefore, most implementations of image-based search vary in their implementation of this feature extraction or implement different methods of comparing the feature vectors to determine similarity. In previous implementations, the features that are extracted from the images in the database are usually stored alongside the raw image data. These feature encodings often take the shape of an n-dimensional vector. To compare the feature vector of a query image to the stored feature vectors, different algorithms can be used to compare the vectors. Such algorithms determine a distance metric, revealing which images are closest in similarity. Examples of such distance metrics are Euclidian distance, Hamming distance, and cosine distance. Several techniques have been proposed to encode the features presented in an image. Most fall into one of two categories: handcrafted features or convolutional neural network (CNN) features.

2.1 Handcrafted Features

To extract features from an image, researchers have previously used what are called "handcrafter features". This includes algorithms such as SIFT [4] and its successors (e.g., SURF [5], BRIEF [6], ORB [7], etc.). Although SIFT-based features are rotation and scaling invariant they have some inherent shortcomings that weaken search performance. With these techniques, a lack of contextual cues might cause false matches where images are similar on local features, but not on regional patches or global features. Furthermore, the lack of a self-training process results in a low discriminative ability. The algorithms also fail on objects with smooth surfaces where there are fewer features to detect. These algorithms have been valuable tools in the early years of image retrieval. However, in recent years, the popularity of SIFT-based image features has been overtaken by the use of CNN techniques. These techniques have been proven to outperform the handcrafted image features [8].

2.2 Convolutional Neural Network Features

In recent years, the limited representation ability of handcrafted features led to the use of more advanced techniques for feature extraction using convolutional neural networks. CNNs are a class of Artificial Neural Networks (ANN) introduced by Lecun et al. [9].

Multiple building blocks form a convolutional neural network, such as convolution layers, pooling layers, and fully connected layers. This technique was developed to adaptively learn spatial hierarchies of features automatically through a backpropagation algorithm. CNNs can represent rich information in a feature vector as compared to handcrafted feature representation. The convolution and pooling layers perform this feature extraction. With each layer, the CNN increases in its complexity, identifying a higher level of the image. Earlier layers focus on simple features, such as colors and edges. As the image data progresses through the layers of the CNN, it starts to recognize larger elements or shapes of the object until it is finally capable of identifying the intended object.

In [10] Babenko et al. proposed a finetuned CNN feature extraction to produce compact feature embeddings to enable fast similarity computation in the filtering step. This approach, however, lacked a description of the details of the image, important for image retrieval. Sharif et al. [11], Gong et al. [12], and Mopuri et al. [13] all proposed various modified versions of CNN for image feature extraction.

Many other models have been designed (e.g., YOLO [14–17], SDD [18], VGG [19], RCNN [20–24], SqueezeNet [25], DenseNet [26], MobileNet [27–29], EfficientNet [30,31] CSPNet [32], etc.) for various computer vision tasks. Mostly image classification, object detection or segmentation. However, these models can also be used for feature extraction. A downside to training CNN for feature extraction is that these models require large amounts of data to train properly. That is why, in absence of a large image dataset, the concept of transfer learning [33,34] is applied. With this method, a CNN trained on a large-scale benchmark dataset, such as ImageNet [35] can be used to learn the features of a new dataset. CNNs pre-trained on large datasets have even been shown to learn feature representations that are generic enough to be applied to tasks they were not trained for, and for such can be used as generic feature extractors [36]. This eliminates the need to train a model specifically on a dataset which is used for image-based similarity search.

2.3 Object Localization

When extracting features from an image, all information in the image is included in the feature vector. This includes backgrounds and other noise that is not relevant when comparing the similarity of products. A way to remove noise from the feature extraction is to localize the object of interest in the image. A variety of localization techniques and applications have been proposed over the years, which include objection detection, extraction, and object segmentation [37,38].

Cinbis et al. [39] used weakly supervised learning where they combined multiple instance learning with CNN-based features to localize objects. In [40] Bazzani et al. proposed a self-taught object localization technique which masks out regions of interest in an image. However, these approaches work on the specific datasets but still leave room for improvement to be used in image-based search systems where datasets are more sensitive to even minor errors in object localization. Any part of an object lost during localization could lead to incomplete feature encodings and ultimately result in incorrect search results.

2.4 Existing Image-Based Search Solutions

Many large e-commerce companies already use image-based search systems. For example, in [41] eBay introduced their image-based search system based on ResNet-50 feature vectors, applying deep semantic binary hashing reaching. They reached a top-1 validation loss of 25.7% on their proprietary dataset. Similarly, in [42] Walmart introduced their image-based search system called SIR. Based on VGG-16 instead of ResNet-50 and using locally sensitive hashing to binarize the extracted feature vectors using projections. Combining these feature vectors with the Elasticsearch engine to retrieve approximate nearest neighbors. They reached a Mean R-Precision of 0.827 on their proprietary product database. This Mean R-Precision metric is computed as the mean of a series of top R relevant images to a query image returned by the system. In [43] researchers at Alibaba introduced a deep CNN approach with modules for category prediction, joint detection and feature extraction, and indexing and retrieval. They reached a 91.01% accuracy on Top-1 category predictions and 62.9% identical recall using the joined detection and feature extraction module at 20 results. Identical recall in this instance refers to cases where there is at least one result belonging to the queried item represented in the query image among the top K retrieved results.

3 Dataset

To the extent of our knowledge, no benchmark for image-based search in domain-specific datasets is available. Therefore, this work uses a custom dataset representing products from the domain of industrial cosmetic packaging. A dataset was created containing images from various domain-specific online platforms. The custom dataset is sorted into generic categories with low intraclass variance and small changes in design and shape are considered different products. The categories of the dataset consist of different types of containers, applicators, and closures. More specifically, there are seven categories of containers: bottle, compact, jar, palette, powder, roll-on, and stick. These items are made to contain cosmetics products. Then there are two categories of applicators: puff and spatula. These items can be used to extract product from a container and apply it to a surface. Finally, there are six categories of closures: general, one-touch, press, dropper, sharp and pump. These items close off containers by screwing or pressing them on the opening. Some can also be used to extract portions of the product from the containers. The overall dataset consists of 2,250 images sorted into 15 categories, where each category contains 150 images. The dataset is divided into training and validation sets with an 80%–20% train-validation split ratio. The training set consists of 1800 images and the validation set consist of 450 images. Another 360 images were collected for a test set, which is completely independent of the training and validation sets. In addition, data augmentation techniques are applied. Using rotation, horizontal flip and zoom to enhance the size of the training dataset.

The original dataset images have various image formats and pixel dimensions. To normalize the images for processing in the proposed image-based search system the images are resized to 224 × 224 pixels and converted to a single channel greyscale format to exclude color from influencing the similarity calculations. However, to remain compatible with the input layers of the pre-trained models in the pipeline the image color dimensions are expanded back to three channels. Fig. 1 shows randomly selected pre-processed sample images from each category in the dataset.

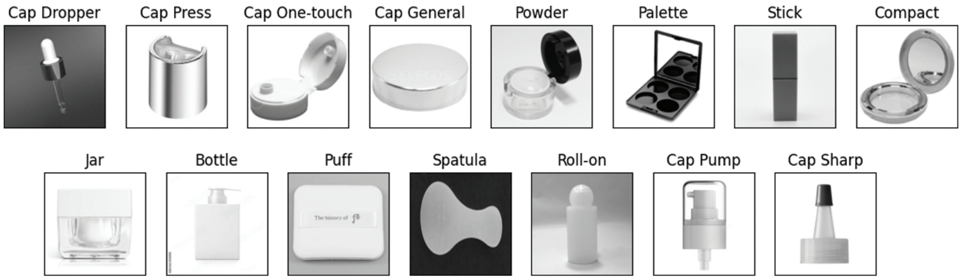

Figure 1: Sample images from each of the 15 categories of the pre-processed cosmetics packaging dataset

4 Proposed Solution

4.1 Object Localization

The background of an image can introduce various amounts of noise to the feature representation. To handle this, the object in the image should be localized and extracted from the background as cleanly as possible, while maintaining a fast-processing time. Hence, a simple yet efficient segmentation network is required. This is accomplished with the U^2-Net architecture [44] pre-trained on the ECSSD dataset [45]. U^2-Net is used because it can process relatively high-resolution images at low computational and memory costs. U^2-Net is comprised of ReSidual U-blocks (RSUs) in the bottom layers and maintains the feature map resolution during multi-scale feature extraction. The network ends in a saliency map fused with an encoder-decoder. To improve the performance on edges an improved FBA matting module is fused to the last output layer of the U^2-Net network. In previous work, FBA matting was used with basic U-Net [46] style architectures. While maintaining a single encoder-decoder to predict the foreground and background, ResNet-50 pre-trained on the ImageNet dataset is also employed to improve FBA matting itself. ResNet building blocks were modified into a bottleneck design by removing strides from two layers of each block.

4.2 Category Prediction

To help narrow down the search and find the most relevant category for the query image a classification model is proposed. The classification model uses the EfficientNetV2L architecture with weights pre-trained on ImageNet as it outperformed previous models in both training speed and parameter efficiency [31]. It has progressive learning, which adaptively adjusts regularization along with image size.

To improve the classification model learning, several parameters in optimization, layers and callbacks are introduced. The fully connected layer at the top of the network is excluded while loading the model. Removing the final dense layers allows for defining a different input size since the output size will increase or decrease accordingly. A new dense layer is added of 1024 neurons with a ReLU activation function. This layer is added so that the model can learn more complex functions and improves classification results. After several experiments, a dropout layer of 0.5 is added after the first dense layer to avoid overfitting. Then after the dropout, a second dense layer of 512 neurons with a ReLU activation function is added. The final output layer is a dense layer with a softmax activation function, which has an equal number of neurons to the number of classes in the dataset.

For training optimization, different algorithms were tested. Out of RMSprop, Adam and stochastic gradient descent (SGD), the latter had the best training results. SGD was used with a momentum of 0.9, a learning rate of 1e^{-3} and Nesterov's accelerated gradient method, as recommended by Sutskever et al. in [47]. Momentum is a process of accelerating gradient vectors in the correct direction and helps in faster convergence. Nesterov's accelerated gradient method is also used to assist in computing gradients from a point which is directed towards the current momentum.

In the next step, the Keras ReduceLROnPlateau function is used to reduce the learning rate whenever the learning metrics stop improving for more than 5 epochs in a row. Often models see a benefit from a factor 2–10 reduction in the learning rate once learning stagnates. Next to this, a decay rate of 0.5 is added. Learning rate decay is another method to adjust the learning rates automatically during training. With this method, the learning rate decays from the original starting value during training, which assists in generalization and optimization. With SGD, a decaying learning rate helps the network to converge to a local minimum by avoiding oscillation and assists in escaping spurious local minima. Fig. 2 shows an overview of the classification model architecture.

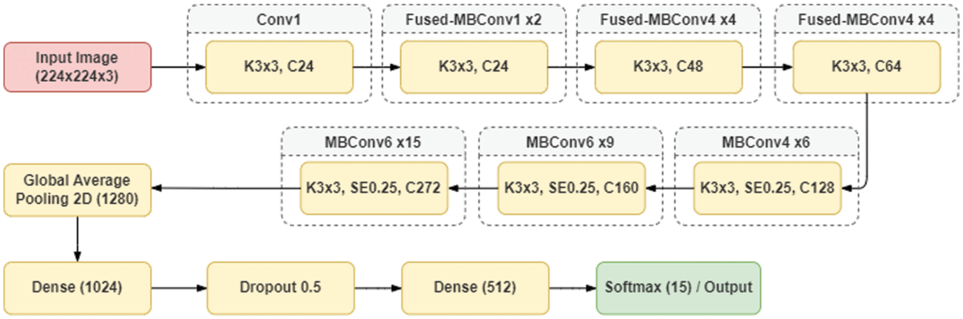

Figure 2: EfficientNetV2L classification model architecture

4.3 Feature Extraction and Similarity Search

To effectively compare the query image to the database images, a ResNet-50 model with weights pre-trained on ImageNet is used to generate feature vectors. Fig. 3 shows feature maps on layers 13 and 17 of the ResNet-50 model. The feature vectors are compared using a k-nearest neighbor (KNN) algorithm which uses the Euclidian distance metric and a brute-force comparison method. The brute-force method means that every feature vector in the database is compared to the feature vector of the query image. Therefore, finding the nearest neighbors of the query feature vector in the vector space of the database has a time complexity of [O(n2)], where n is the number of feature vectors in the database. This time complexity is reduced by predicting the category of the query image and limiting the search to the predicted category.

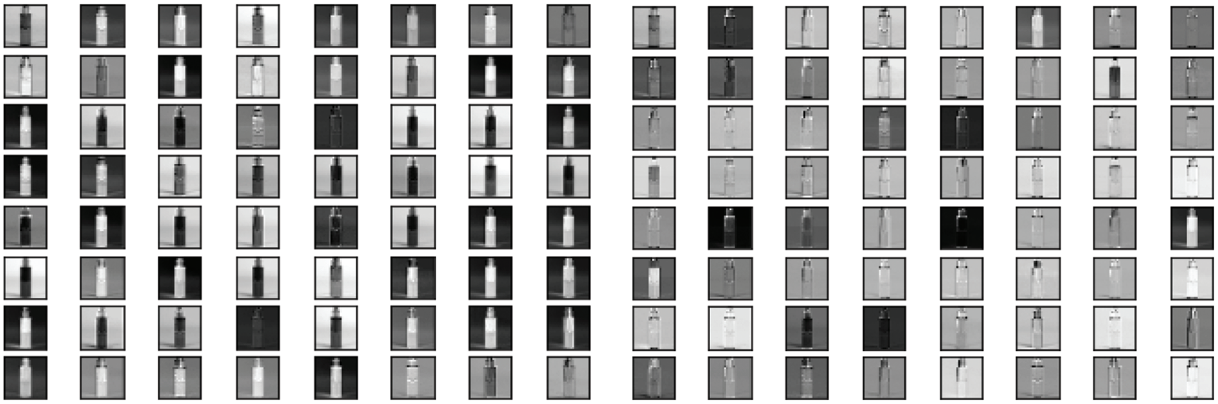

Figure 3: ResNet-50 feature maps after layer 13 (left) and layer 17 (right)

The Euclidian distance is used to compare each vector and find the nearest neighbors. The function for the Euclidian distance in an n-dimensional space is denoted as Eq. (1), considering two n-dimensional points $(p_1, \ldots p_i, \ldots p_n)$ and $(q_1, \ldots q_i, \ldots q_n)$, where n is the number of dimensions of the feature vectors, which in the proposed solution is 2048.

$$d\left(p, q\right) = \sqrt{\left(p_1 - q_1\right)^2 + \left(p_2 - q_2\right)^2 + \cdots + \left(p_i - q_i\right)^2 + \cdots + \left(p_n - q_n\right)^2} \tag{1}$$

Intersection over Union (IoU) is applied to the top results of the KNN search. The IoU score is calculated on the contour of the images. This method is proposed to favour the shape of the objects more strongly in the search results. The images with the highest IoU score compared to the query image are the most similar in shape and size. Fig. 4 shows a visualization of IoU by overlaying a query image with several retrieved images. The contour is created as an array of Boolean values generated from the image where the object is localized and the background pixels are omitted. From this image, an array is created where for each pixel position where the value of the pixel is white (or empty) a Boolean is set to False and for every other value, a Boolean is set to True. This transformation function can be denoted as Eq. (2), where $f(x, y)$ is the input image on which the transformation is performed and $g(x, y)$ the transformed image.

$$g(x, y) = \begin{cases} 0 & f(x, y) = 255 \\ 1 & f(x, y) < 255 \end{cases} \tag{2}$$

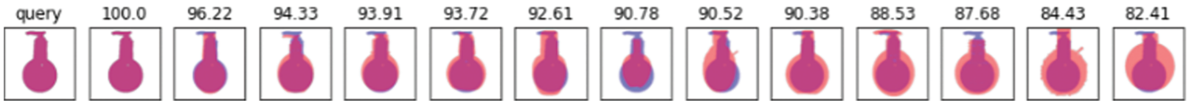

Figure 4: Visualizations of the intersection of the contour of a query image with the contours of its top retrieved results ordered by IoU score based on object shape size

4.4 Online/Offline Pipeline

The final image similarity search pipeline consists of both an offline and an online stage. In the offline stage information from the images in the database is extracted and saved alongside the original image files. In the online stage, the information of a query image is extracted to then be compared to the already extracted information of the database images. The information extracted from the images consists of three parts: class, feature vector, and contour. The information extracted from the query image that comes in at the online stage of the pipeline is compared to the information extracted from the database images at the offline stage. First, the search is narrowed down using classification to choose a category to search from in the database. Then, to calculate the overall similarity of the images, the feature vector of the query image is compared to the feature vectors in the database, using the KNN algorithm with the brute-force method and the Euclidian distance metric. Finally, the contour of the query image is compared to the contours in the database using the IoU score to re-rank the KNN search results. Fig. 5 shows the architecture of the final image similarity search pipeline.

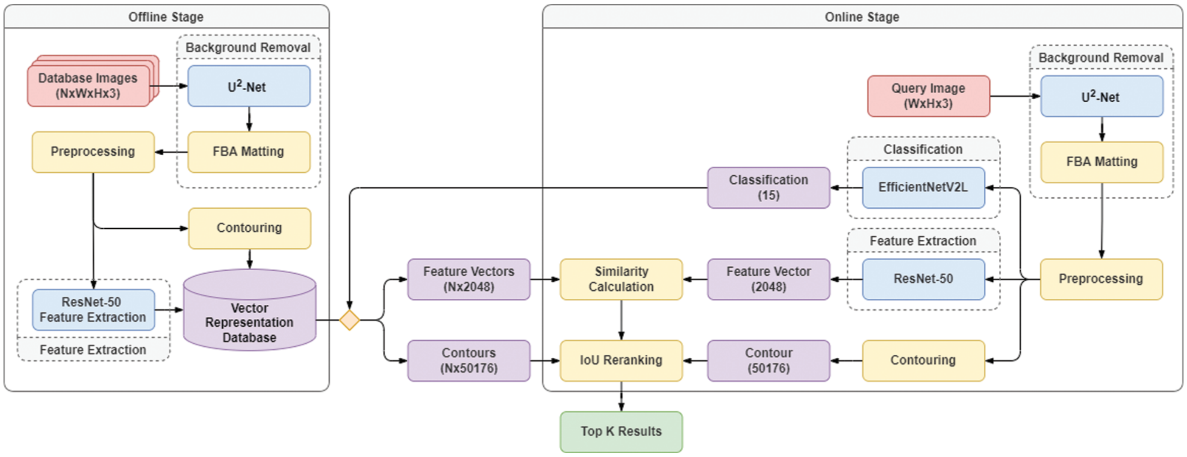

Figure 5: Overview of the entire proposed image similarity search pipeline architecture

5 Experimental Results

In this section, experiments, results, and comparisons with other previous techniques are discussed for all the separate parts and the whole of our proposed solution.

5.1 *Object Localization*

In our proposed approach the removal of background noise from the feature embeddings plays a crucial part. Fig. 6 shows a comparison of different image localization techniques which were evaluated to find the algorithm which extracts the object in the image efficiently without losing any key features or retaining noise. Orange boxes show the bounding boxes of the localized objects as predicted by the model. Red dotted boxes show missing parts of objects after localization. The first row shows examples of object detection with the SSD [18] algorithm. This row exemplifies that with this algorithm the detected bounding boxes do not always completely localize all key parts of the object in the image. The second row shows an example of object detection with YOLOv4 [17], which might also fail in some cases to extract all key parts of an object entirely. These models localize the object in the images and return bounding box coordinates which might clip off important parts of the object. Another problem with these techniques is that the detected bounding box still includes parts of the background, which still incorporate noise into the feature embedding. Some parts of an object not being fully detected could contribute to losing valuable information during feature extraction. For this reason, both these techniques do not perform well for detailed image-based similarity search. For example, if the goal is to search for a bottle with a pump cap and the pump cap is clipped off during the detection process the system will return results without pump caps. The third row of the image shows examples of results of the BASNet [48] segmentation algorithm. Instead of returning bounding box coordinates this algorithm segments out the object of interest, discarding background information. Its performance is better than SSD or YOLOv4 but during segmentation still loses some part of the detected object. Finally, the fourth row shows the object localization results of our proposed model. This approach can effectively extract the objects from images without significant noise or loss of key parts of the detected object.

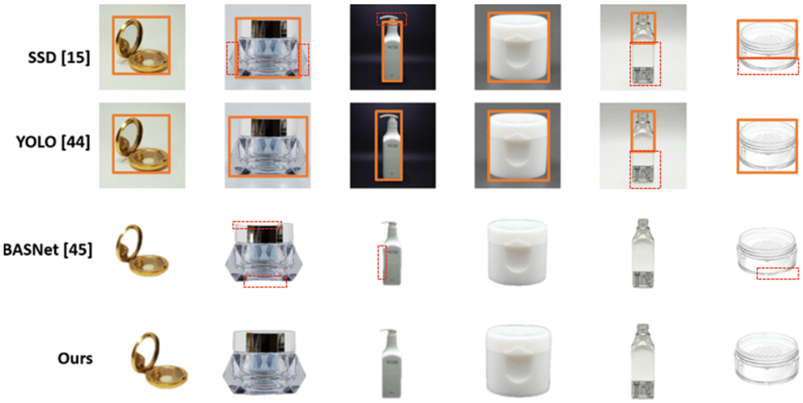

Figure 6: Comparison of different methods for object localization and removing backgrounds

5.2 *Category Prediction*

For category prediction, extensive experiments were performed, and comparisons were made of various CNN models, including a hyperparameters sweep using various optimization algorithms. With all the models several optimizers and learning rates were tested in combination with learning rate decay and early stopping when the learning metrics stop improving for more than 10 epochs. The results and optimizer tests for the ResNet101V2 and MobileNetV2 models can be found in suppl. 1 showing the performance of several hyperparameter and epochs. In these tests the ResNet101V2 model achieved a training accuracy of 99% and a validation accuracy of 90% in 37 epochs. While the MobileNetV2 model achieved a training accuracy of 99% and a validation accuracy of 92% in 44 epochs.

Our proposed model achieved a training accuracy of 97% and a validation accuracy of 94% in 76 epochs. Fig. 7 shows the accuracy and loss for the training and validation datasets during training. Table 1 compares the average accuracy of the evaluated classification models on the completely independent test dataset. The table shows the metrics precision (3), recall (4) and the F1-score (5) for the proposed model [49]. These metrics can be denoted in the following equations, where TP = True Positive, FP = False Positive and FN = False Negative.

$$precision = TP/(TP + FP) \tag{3}$$

$$recall = TP/(TP + FN) \tag{4}$$

$$F1{-}score = 2/(1/precision + 1/recall) \tag{5}$$

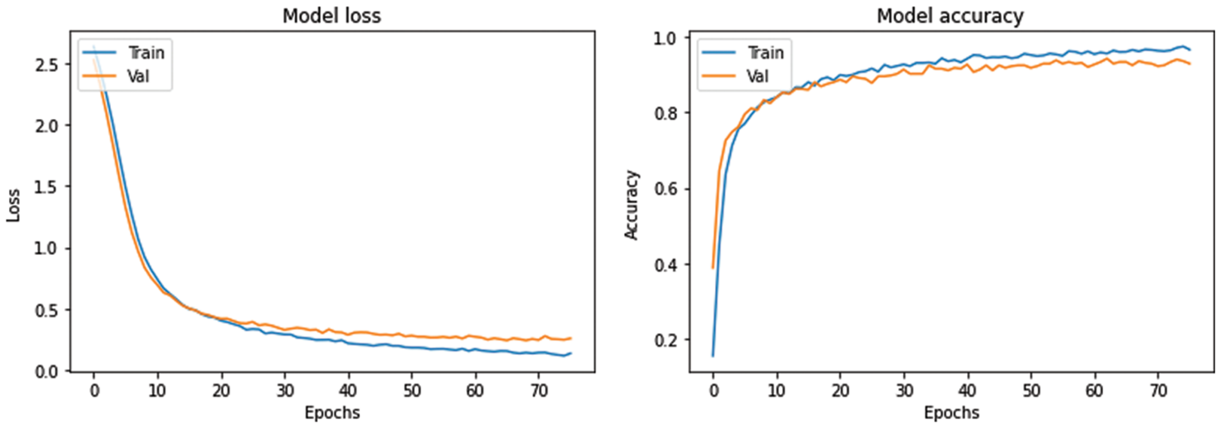

Figure 7: EfficientNetV2L classification model loss and accuracy during training

Table 1: EfficientNetV2L classification report

Class	Precision	Recall	F1-score
Bottle	0.92	0.92	0.92
Jar	1.00	0.96	0.98
Compact	0.96	0.92	0.94
Palette	1.00	1.00	1.00
Powder	0.88	0.96	0.92
Roll-on	1.00	0.96	0.98
Stick	0.96	0.96	0.96
Cap dropper	1.00	1.00	1.00
Cap general	0.92	0.92	0.92

(Continued)

Table 1: Continued

Class	Precision	Recall	F1-score
Cap one-touch	1.00	0.96	0.98
Cap press	0.92	1.00	0.96
Cap pump	1.00	1.00	1.00
Cap sharp	1.00	1.00	1.00
Puff	0.96	0.96	0.96
Spatula	1.00	1.00	1.00

Furthermore, the accuracy of the model per class can be denoted as Eq. (6), where TP = True Positive, FP = False Positive, TN = True Negative and FN = False Negative.

$$accuracy = (TP + TN)/(TP + TN + FP + FN) \tag{6}$$

The average accuracy of the model is then calculated by taking the average of the per class accuracy.

Our proposed model achieved the highest accuracy and the figures in suppl. 1 show that the other models are overfitting even before training for the same number of epochs as our proposed model. Finally, when calculating the variance in the per-class accuracy our model has the lowest variance and the minimum per-class accuracy is the highest. In Table 2, the performance of our proposed model is shown for every one of the 15 classes in the test dataset. Overall, averaged over the 15 classes, the model reached an accuracy of 97%. Fig. 8 shows the confusion matrix of the classification model.

Table 2: Classification model comparison

Model	Parameters	Average accuracy
EfficientNetV2L	119.6 M	97%
MobileNetV2	4.1 M	96%
ResNet101V2	45.3 M	95%

5.3 Qualitative Image-Based Search Results

Fig. 9 shows some qualitative results of the final image similarity search pipeline. Every row on either side of the image shows a query image from the test dataset followed by the top 5 results the pipeline returned. An additional qualitative comparison between a classic image-based search implementation and our approach can be found in suppl. 2. With our dataset and hardware, a search query on average took 0.2 s. A Linux sever containing an Nvidia RTX 2080 Ti GPU with 11 GB of VRAM was used to train and test the proposed solution.

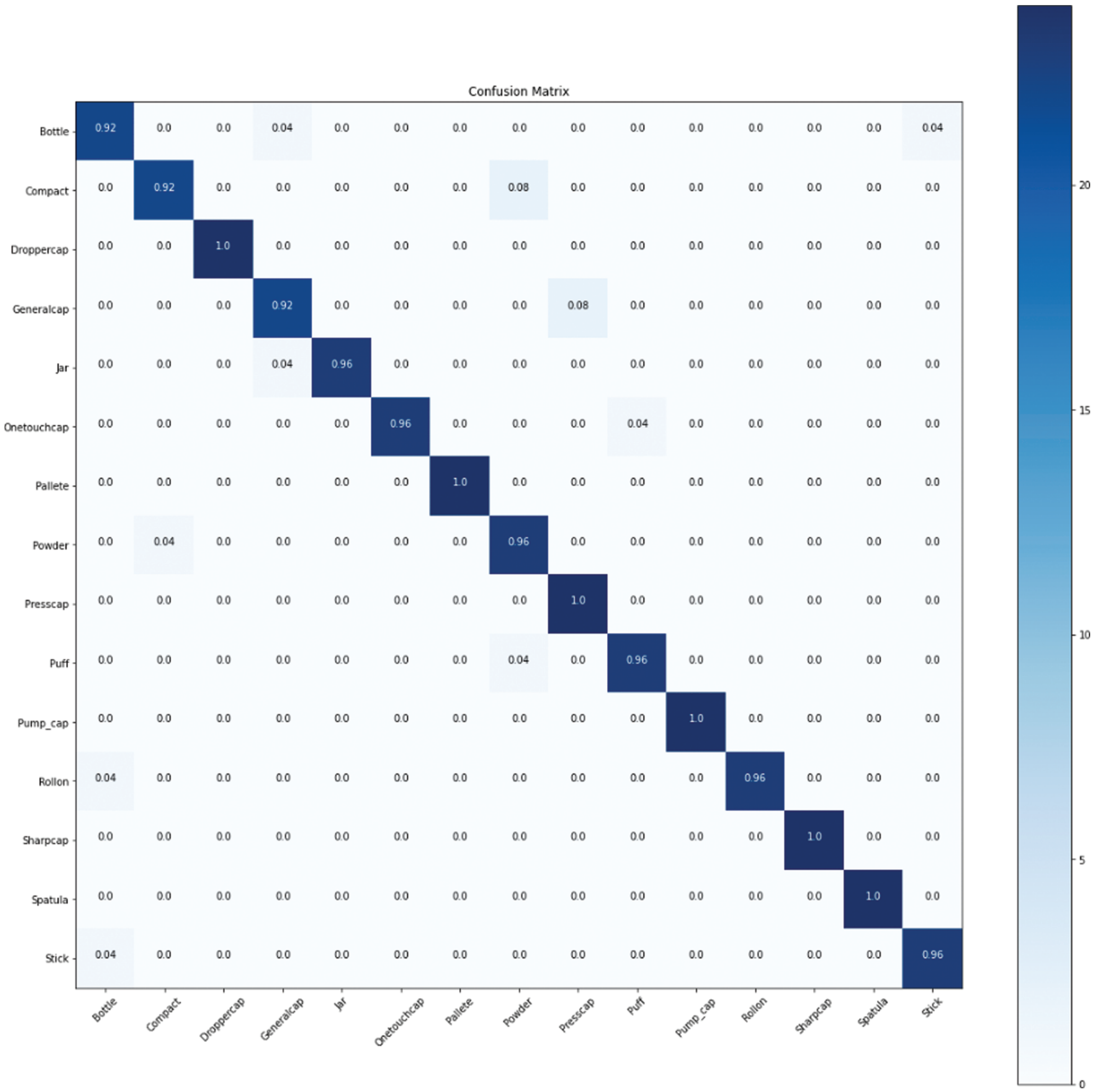

Figure 8: Confusion matrix of the proposed category classification model

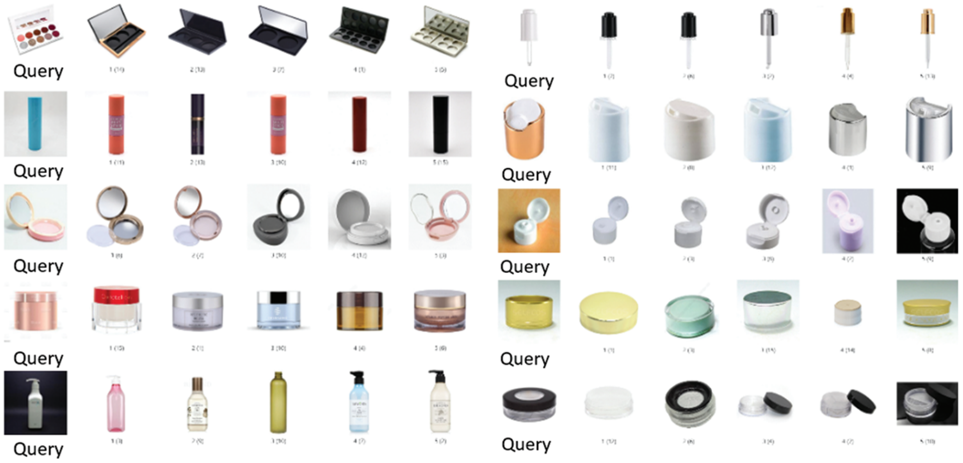

Figure 9: Top 5 image similarity search results for 10 sample query images from the test set. The first image in each row on each side presents the query followed by five results

6 Conclusion

In this paper, a multimodal fused deep learning network for image-based search is proposed to reduce the gap between expected and retrieved images in sensitive domain-specific datasets. Due to nonhomogeneous image properties and diversity in image datasets, general image search engines cannot be employed for domain-specific datasets. Especially for products or objects which need to be retrieved with a high precision of shape or design with extremely small variance needs very accurate feature representation. The proposed pipeline uses an improved object localization technique to extract objects without losing essential object parts and reducing noise for feature extraction and representation. To reduce the time complexity and to retrieve more optimal results a classification model is proposed, which reduces the search scope in large datasets. Furthermore, the addition of the IoU metric on top of the combined similarity search with brute and Euclidean distance improved the retrieved results in cases where the searched items are very similar in shape or design but different in size. The proposed pipeline provides optimal results as compared to general image search engines or comparative approaches. Quantitative and qualitative experiment results showed improved ranking of retrieved results based on the proposed method.

In future work, the construction of feature vectors should be improved to further extract extremely minor details of objects of interest. This is crucial for cases where images already look particularly similar and only have minute differences. This could also enhance the classification performance on large datasets with a massive number of classes and further improve the efficiency and robustness of the proposed scheme. A method should also be found to reduce the influence of image perspective, where the angle from where an image is taken will significantly influence search results.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] TinEye, 2008. [Online]. Available: https://www.tineye.com.

[2] Google images, 2001. [Online]. Available: https://images.google.com.

[3] Bing image search, 2009. [Online]. Available: https://www.bing.com/images.

[4] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. of the Seventh IEEE Int. Conf. on Computer Vision*, Kerkyra, Greece, vol. 2, pp. 1150–1157, 1999.

[5] 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, 2008.

[6] M. Calonder, V. Lepetit, C. Strecha and P. Fua, "BRIEF: Binary robust independent elementary features," in *European Conf. of Computer Vision–ECCV 2010. Lecture Notes in Computer Science (LNCS)*, Crete, Greece, vol. 6314, pp. 778–792, 2010.

[7] E. Rublee, V. Rabaud, K. Konolige and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *2011 IEEE Int. Conf. on Computer Vision*, Barcelona, Spain, pp. 2564–2571, 2011.

[8] L. Zheng, Y. Yang and Q. Tian, "SIFT meets CNN: A decade survey of instance retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Computer Society*, vol. 40, no. 5, pp. 1224–1244, 2018.

[9] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[10] A. Babenko, A. Slesarev, A. Chigorin and V. Lempitsky, "Neural codes for image retrieval," in *European Conf. of Computer Vision–ECCV 2014. Lecture Notes in Computer Science*, Zurich, Switzerland, vol. 8689, no. 1, pp. 584–599, 2014.

[11] A. Sharif, R. H. Azizpour, J. Sullivan and S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition," in *2014 IEEE Conf. on Computer Vision and Pattern Recognition Workshop (CVPRW)*, Columbus, OH, USA, pp. 512–519, 2014.

[12] Y. Gong, L. Wang, R. Guo and S. Lazebnik, "Multi-scale orderless pooling of deep convolutional activation features," in *European Conf. of Computer Vision–ECCV 2014. Lecture Notes in Computer Science*, Zurich, Switzerland, vol. 8695, pp. 392–407, 2014.

[13] K. R. Mopuri and R. Venkatesh Babu, "Object level deep feature pooling for compact image representation," in *2015 IEEE Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Boston, MA, USA, pp. 62–70, 2015.

[14] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Vegas, NV, USA, pp. 779–788, 2016.

[15] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, vol. 2017-January, pp. 6517–6525, 2017.

[16] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv: Computer Vision and Pattern Recognition*, 2018.

[17] A. Bochkovskiy, C. -Y. Wang and H. -Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," *arXiv: Computer Vision and Pattern Recognition*, 2020.

[18] L. Wei, D. Anguelov, D. Erhan, C. Szegedy, S. Reed *et al.,* "SSD: Single shot MultiBox detector," in *European Conf. of Computer Vision–ECCV 2016. Lecture Notes in Computer Science*, Amsterdam, The Netherlands, vol. 9905, pp. 21–37, 2016.

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv: Computer Vision and Pattern Recognition*, 2015.

[20] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Columbus, OH, USA, pp. 580–587, 2014.

[21] R. Girshick, "Fast R-CNN," in *2015 IEEE Int. Conf. on Computer Vision (ICCV)*, Santiago, Chile, pp. 1440–1448, 2015.

[22] A. Salvador, X. Giro-i-Nieto, F. Marques and S. Satoh, “Faster R-CNN features for instance search,” in *2016 IEEE Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Vegas, NV, USA, pp. 394–401, 2016.

[23] J. Dai, Y. Li, K. He and J. Sun, “R-FCN: Object detection via region-based fully convolutional networks,” *arXiv: Computer Vision and Pattern Recognition*, 2016.

[24] J. Pang, K. Chen, J. Shi, H. Feng and W. Ouyang, “Libra R-CNN: Towards balanced learning for object detection,” in *2019 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, pp. 821–830, 2019.

[25] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally *et al.,* “SqueezeNet: AlexNet-level accuracy with 50× fewer parameters and <0.5 MB model size,” *arXiv: Computer Vision and Pattern Recognition*, 2016.

[26] G. Huang, Z. Liu, L. van der Maaten and K. Q. Weinberger, “Densely connected convolutional networks,” in *2017 30th IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, vol. 2017-January, pp. 2261–2269, 2017.

[27] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang *et al.,* “MobileNets: Efficient convolutional neural networks for mobile vision applications,” *arXiv: Computer Vision and Pattern Recognition*, 2017.

[28] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. -C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” in *2018 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, pp. 4510–4520, 2018.

[29] A. Howard, M. Sandler, G. Chu, L. -C. Chen, B. Chen *et al.,* “Searching for mobilenetv3,” in *IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, Seoul, South Korea, pp. 1314–1324, 2019.

[30] M. Tan and Q. v. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” in *36th Int. Conf. on Machine Learning, ICML 2019*, Long Beach, CA, USA, vol. 2019-June, pp. 10691–10700, 2019.

[31] M. Tan and Q. v. Le, “Efficientnetv2: Smaller models and faster training,” in *Int. Conf. on Machine Learning, Online*, 2021.

[32] C. -Y. Wang, H. -Y. M. Liao, Y. -H. Wu, P. -Y. Chen, J. -W. Hsieh *et al.,* “CSPNet: A New backbone that can enhance learning capability of CNN,” in *2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Seattle, WA, USA, pp. 1571–1580, 2020.

[33] M. Oquab, L. Bottou, I. Laptev and J. Sivic, “Learning and transferring Mid-level image representations using convolutional neural networks,” in *2014 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Columbus, OH, USA, pp. 1717–1724, 2014.

[34] H. Liang, W. Fu and F. Yi, “A survey of recent advances in transfer learning,” in *2019 IEEE 19th Int. Conf. on Communication Technology (ICCT)*, Xi’an, China, pp. 1516–1523, 2019.

[35] O. Russakovsky, J. Denk, H. Su, J. Krause, S. Satheesh *et al.,* “ImageNet large scale visual recognition challenge,” *Int. J. Comput Vis*, vol. 115, no. 3, pp. 211–252, 2015.

[36] L. Hertel, E. Barth, T. Kaster and T. Martinetz, “Deep convolutional neural networks as generic feature extractors,” in *Proc. of the Int. Joint Conf. on Neural Networks*, Killarney, Ireland, vol. 2015-September, pp. 1–4, 2015.

[37] A. Jafar, M. T. Hameed, N. Akram, U. Waqas, H. S. Kim *et al.,* “CardioNet: Automatic semantic segmentation to calculate the cardiothoracic ratio for cardiomegaly and other chest diseases,” *Journal of Personalized Medicine*, vol. 12, no. 6, pp. 988, 2022.

[38] R. Ali Naqvi, D. Hussain and W. -K. Loh, “Artificial intelligence-based semantic segmentation of ocular regions for biometrics and healthcare applications,” *Computers, Materials & Continua*, vol. 66, no. 1, pp. 715–732, 2020.

[39] R. G. Cinbis, J. Verbeek and C. Schmid, “Weakly supervised object localization with multi-fold multiple instance learning,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 1, pp. 189–203, 2017.

[40] L. Bazzani, A. Bergamo, D. Anguelov and L. Torresani, “Self-taught object localization with deep networks,” in *2016 IEEE Winter Conf. on Applications of Computer Vision (WACV)*, Lake Placid, NY, USA, pp. 1–9, 2016.

[41] F. Yang, A. Kale, Y. Bubnov, L. Stein, Q. Wang *et al.,* “Visual search at eBay,” in *KDD ’17: 23rd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, Halifax NS Canada, pp. 2101–2110, 2017.

[42] T. Stanley, N. Vanjara, Y. Pan, E. Pirogova, S. Chakraborty *et al.,* “SIR: Similar image retrieval for product search in E-commerce,” in *Similarity Search and Applications: 13th Int. Conf., SISAP 2020*, Copenhagen, Denmark, pp. 338–351, 2020.

[43] Y. Zhang, P. Pan, Y. Zheng, K. Zhao, Y. Zhang *et al.,* “Visual search at alibaba,” in *KDD ’18: 24th ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining*, London United Kingdom, vol. 18, pp. 993–1001, 2018.

[44] X. Qin, Z. Zhang, C. Huang, M. Dehghan, O. R. Zaiane *et al.,* “U 2-net: Going deeper with nested U-structure for salient object detection,” *Pattern Recognition*, vol. 106, pp. 107404, 2020.

[45] J. Shi, Q. Yan, L. Xu and J. Jia, “Hierarchical image saliency detection on extended CSSD,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 4, pp. 717–729, 2016.

[46] O. Ronneberger, P. Fischer and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015. MICCAI 2015. Lecture Notes in Computer Science (LNCS)*, Munich, Germany, vol. 9351, pp. 234–241, 2015.

[47] I. Sutskever, J. Martens, G. Dahl and G. Hinton, “On the importance of initialization and momentum in deep learning,” in *30th Int. Conf. on Machine Learning 2013*, Atlanta, GA, USA, vol. 28, no. 3, pp. 1139–1147, 2013.

[48] X. Qin, D. -P. Fan, C. Huang, C. Diagne, Z. Zhang *et al.,* “Boundary-aware segmentation network for mobile and web applications,” *arXiv: Computer Vision and Pattern Recognition*, 2021.

[49] A. Tharwat, “Classification assessment methods.” *Applied Computing and Informatics*, vol. 17, no. 1, pp. 168–192, 2021.