Delver Hamed : Content Based Image Retrieval Using BEIT Transformer and Detic

Mostafa Jelveh[1]

[1]Affiliation not available

May 06, 2024

Delver Hamed[1]: Content Based Image Retrieval Using BEIT Transformer and Detic

Mostafa Jelveh
mostafaajelveh@gmail.com

***Abstract*—I present DHam, a new and exact unsupervised learning model for Content Based Image Retrieval (CBIR) that does not need any training data set. DHam is accurate especially when you deal with the multiple objects image with background (MOIB). This is the first time that pre-trained Detic and pre-trained of a self-supervised based image transformer (SSIT) BEIT, have been mixed for CBIR. First, I use pre-trained Detic to detect image objects. Then I extract every object's feature with pre-trained BEIT. DHam shows its superiority when search is accomplished amongst multiple objects images with background (MOIBs). Besides, DHam Test results are compared with pure BEIT and pure ResNet CBIR models. On the other hand, it is not a fast model. It takes around 19 and 273 seconds to compare the input image with 44,891 and 1,868,672 features respectively. Compared to state of the art CBIR systems, DHam may bring irrelevant images but is less likely to miss the target similar image.**

***Index Terms*— BEIT, Transformer, CBIR, DHam, Detic, multiple objects image with background (MOIB), Self-Supervised Based Image Transformer (SSIT).**

I. INTRODUCTION

CBIR's usage is becoming popular in many criteria from architectural and engineering design to intellectual property and it's demand is dramatically increasing in recent years. Besides, the volume of image data has spiked all over the digital environment from internet to intranet, which inevitably requires a strong searching system based on visual properties.

In real world, we may not deal with clean, single object images in sample space dataset. We may have multiple objects and backgrounds in an image. In addition, MOIBs are common in raw image data. It is CBIR's network capability to find the right images according to query image.

Several CBIR ground truth datasets have one object in an image. They have multiple views of one object in number of files like namely Stanford Online, CUB200, In-Shop dataset [6], UK Bench dataset [7] and COIL 20, COIL 100 [11]. Some others search a sole visual definition in an image like INRIA holidays, Paris6K, Oxford5K [7] and Corel1k dataset in [4]. If the model guesses the image notion and brings the same concept images, the model is successful. DHam does not look at the image in a sole concept. It divides the image into several parts regarding its number of objects.

Firstly, I went one step further and selected a MOIB dataset. I used COCO Val 2017[17] that contains 5000 crowded images that does have images with multiple objects with background. Secondly, using my system, in addition to bringing the same concept images, the model searches all the image objects one by one in every image to detect the most similar shape objects in images. I used an individual metric for the model measurement. This metric expects the system to find the defined sole exact target image instead of finding the same concept several images. If the system is not able to find the target image, it receives zero score.

There were Active learning algorithms that needed oracle intervention and Interactive response time to label the data in order to update the learning [7].

[9] Mixed dominant color descriptors, (Hue, Saturation and value) and texture features which was gained by wavelet and curvelet features with particle swarm optimization algorithm. [10] Made a model for textual and non-textual images. It used MSER method for textual images sand BRISK method for non-textual images. [11], [12] concentrated Gaussian Hermite moments and Pseudo Zernike moments descriptors respectively as shape features. [11] Introduced a method based on Gaussian Hermite moments and SVM as a shape descriptor. [12] Used first Zernike moments to calculate the query image interval and Pseudo moments for image features. Database images which are out of interval are ignored which causes the model's speed up.

Recent CBIR systems use convolutional layers as deep learning architecture for feature extraction [2], [5], [8], [13].

Some of The most Recent CBIR systems use Transformers for feature extraction that has led to satisfactory results [3], [4], [6], [7], [8]. Transformers have made a big evolution in AI even in image processing.

[1] My Nick name is Hamed.

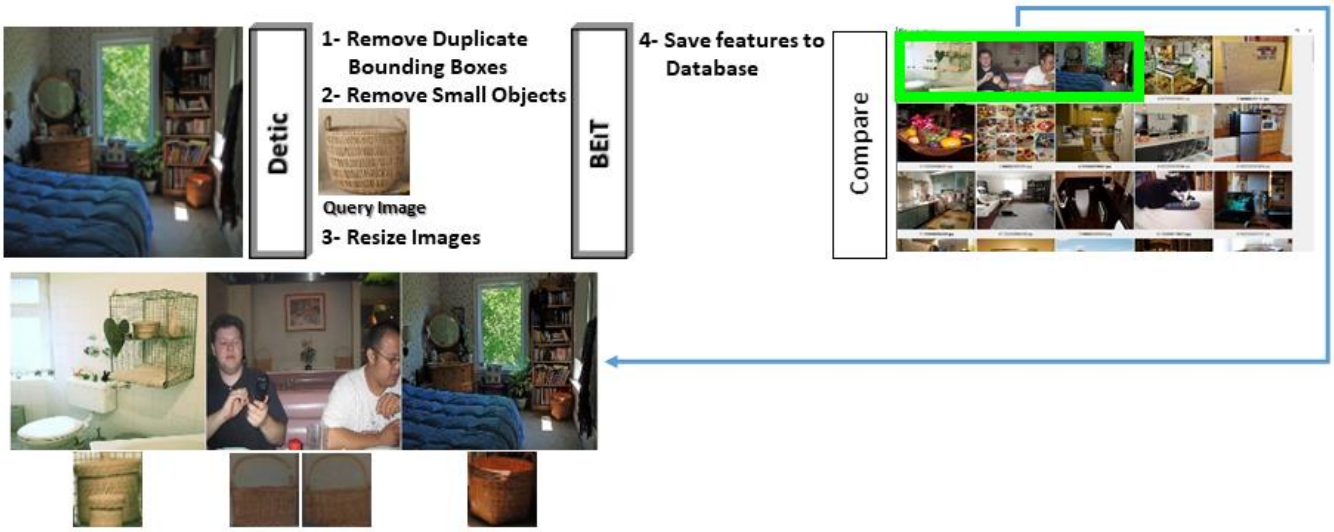

Fig. 1. DHam System. After Object Detection with Detic, my system (1) removes Duplicate Bounding Boxes, (2) removes objects smaller than 1% of image size, (3), Resizes images to 224×224. Then after feature extraction with BEiT, (4) saves it to database. When query image is given, DHam (3) resizes the image to 224 × 224 and then extracts its features with BEiT. Finally system compares query image features with database features.

[4], [6], [7] and [8] used Vision Transformer as a feature extractor.

[6] Used Vison Transformer with metric learning objective.

[8] Used Vision Transformer for Sketched-Real Image Retrieval (SRIR) beside Info-GAN on ESRIR dataset.

I used BEiT Transformer that gained better top1-accuracy than Vision Transformer in ImageNet dataset for classification task [15].

I used and mixed two pre-trained strong transformer's based models BEiT and Detic [14], [15] that already have been trained with massive data. In fact DHam is made for object retrieval in images. DHam is practical, simple, straightforward, easy to implement and outperforms state of the art methods.

DHam detects the image objects with Detic. Detic is a strong transformer based object detection model that can detect 21,841 classes. Then DHam looks at every object of image independently and extract its features with BEiT. The whole image is also looked as a single concept, and the features are extracted from it. All features are stored within the database for subsequent comparison with the features of query image: see Figure 1.

Furthermore, I have developed distinct scoring metrics tailored specifically for evaluating the algorithm's correctness other than accuracy, recall and mAP metrics.

II. Proposed Method and KPI

A. Proposed Method

Detic object detection system may produce duplicate bounding boxes. As my requirement involves solely bounding boxes, I initially eliminate duplicate bounding boxes to mitigate the generation of redundant features following detection.

Often Very Small objects in an image, does not have enough visual quality. Besides, usually we want to detect the main objects in the image, not very small ones. I remove the images that their size are smaller than 1% of the original image. If I have extra features for every image, it affects model's speed In retrieval process and also number of saved features and database inevitably increases.

As you can see in Figure 2, number of all bounding boxes of the images 1 and 2, after duplicate bounding boxes removal are 111 and 36 respectively. Now after removing tiny images they will be 11 and 13, which saves 90% and 64% feature space for the extraction of these two images independently. On the other hand, for image retrieval we have a smaller more efficient database.

Now that we have squeezed number of objects, I resize every cropped object to 224×224 and get its feature with BEiT -L: see Figure 4 that shows whole DHam's mechanism that is a kind of unsupervised learning. Then it is saved in database. Here I transform a single image to multiple images, with the distinction that I refrain from saving the extracted images. Just I save their features in my database. In addition to considering every dataset image as a sole concept, DHam also considers every part of image that an object is guessed, as an individual concept and collects its features. I used Feather file format for database system. Upon presenting the query image to DHam, it proceeds to extract image feathers and subsequently compares the query features with those stored in the database. For similarity search I tested Annoy for DHam [4]. It is fast, but it decreases the DHAM's accuracy and modifies outcomes. Thus, I tested Manhattan, Euclidean and Cosine distance. DHam worked much better with Cosine and I chose this method for DHam. Generally in CBIR we do not want to miss even one similar image because it totally affects to your final decision. Suppose that you are an industrial design (Intellectual Property)

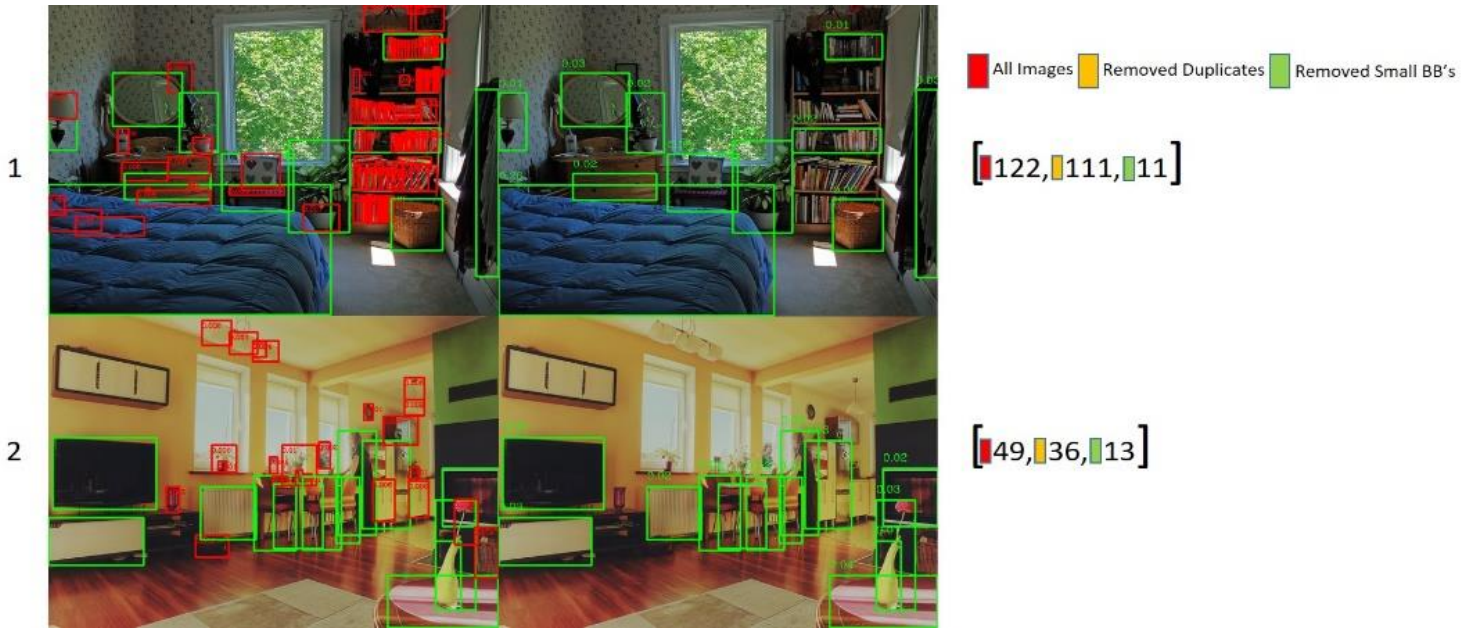

Fig. 2. DHam removes duplicate bounding boxes presented within Detic output. DHam removes the images that their square are less than 1% of the image size.

Examiner and you are searching for an industrial design declaration product image to see it's possible
Similarities or duplicates. If you have one similar image in the Database and model misses to discover it, you probably conclude it is a new and novel design instead of rejecting the declaration. But if the model shows the both unrelated images and the sole similar target image, examiner can ignore the unrelated images and distinct the sole similar image since the results are limited to 200 images in DHam. In other words in most cases in CBIR, False positives (mAP) are less important than False Negatives (Recall). False Positives can be disregarded by human intelligence, but if we have False Negative, how can we find it through thousands of images.
I tested to classify the database and search the query image similarities in related class not in whole database. It increased the mAP and model speed. On the other hand, it could decrease the Recall. As False Negatives were vital for me, I preferred to search the whole database. Overall, my purpose was to make a reliable model.

B. Proposed KPI

DHam brings the first 200 images similar to the query image in similarity descending order. In model testing, for every test query image, I have defined a target image in database that model should be able to find it. For examining the model, for every test image I selected one object from a Multi Object Image with Background (MOIB) in my dataset and called it Target Image. Then I gave a one object image as query image to the model that was similar to Target Object. The Query Object may have a different color, material, size, pose and angle from Target Image: see test images in Figure 6. Optimistically I expect that model finds the target image in the first 10 pictures of the output to get the maximum score.

I use a KPI other than accuracy, mAP and recall here. We call it Target Image Score (TIS): see Figure 3. TIS is zero if target image is not shown in the 200 images list. $TIR_{(X)}$ Is Target image rank of X_{th} image in DHam. If DHam fails to find it within 200 image results, $TIR_{(X)}$ is equal to zero.

$$TIS_{(x)} = 1 - \left(\left[\frac{TIR_{(X)}}{10}\right] \times 0.0.5\right) \qquad (1)$$

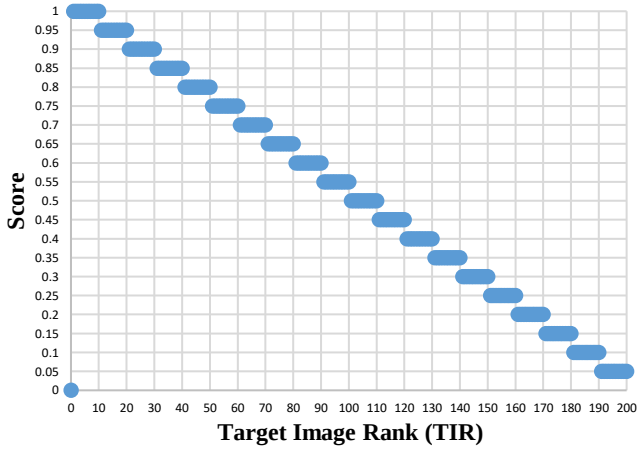

Fig. 3. Target Image Score (TIS)

I give a Difficulty Score (DS) to every MOIB in database that is taken part in testing process. The harder it seems to find the target object according to the query image with human intelligence, I give a higher Difficulty score to the target MOIB. $DS_{(x)}$ Is an integer between 5 and 10: see (2). In other words, it

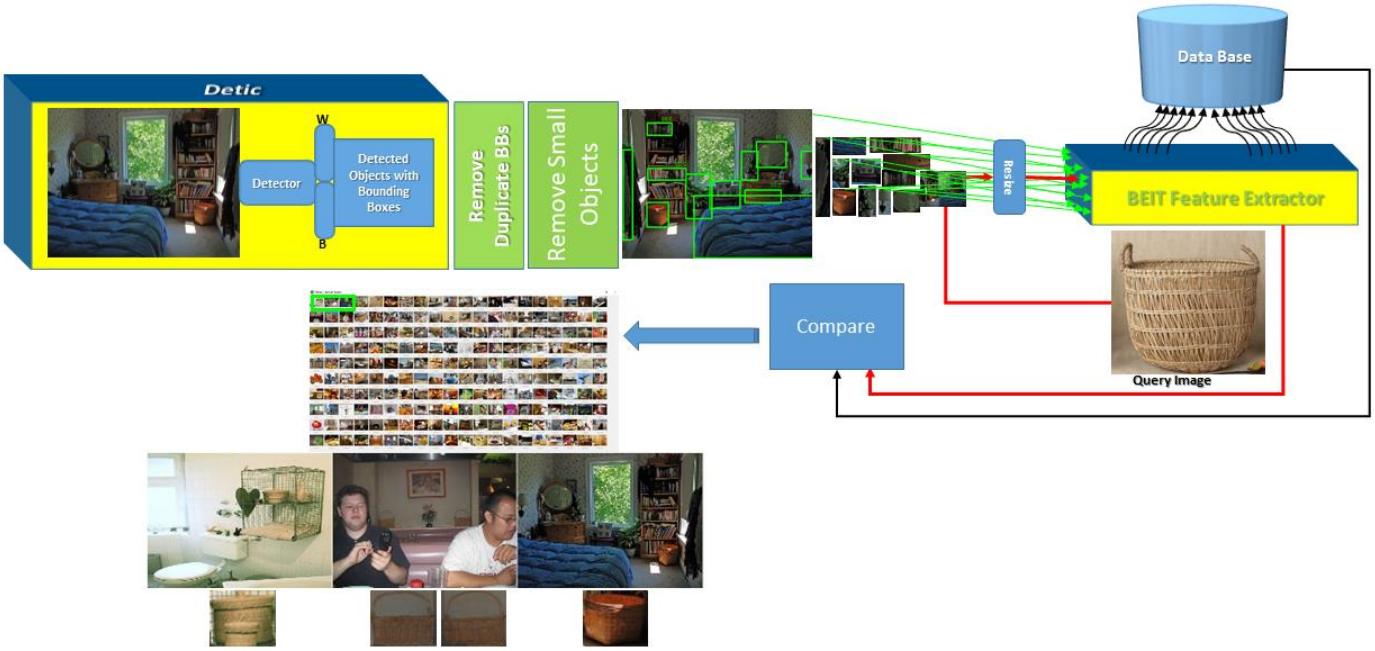

Fig. 4. DHam's System schematic

Is actually an image weight.

$$DS_{(x)} \in (4,10] \ \ int \tag{2}$$

$$Mean\_TIS = \frac{1}{\sum_{i=1}^{n} DS_{(i)}} \sum_{i=1}^{n} TIS_{(i)} \times DS_{(i)} \tag{3}$$

$$f(x) = \begin{cases} 1 \ \ if \ x \neq 0 \\ 0 \ if \ x = 0 \end{cases} \tag{4}$$

$$SR_{W_{tot}} = \frac{1}{\sum_{i=1}^{n} DS_{(i)}} \sum_{i=1}^{n} f(TIR_{(i)}) \times DS_{(i)} \tag{5}$$

$$SR_{tot} = \frac{1}{n} \sum_{i=1}^{n} f(TIR_{(i)}) \tag{6}$$

Average Score (Mean_TIS) is average of image scores with their related weights. I defined a Success Rate (SR) for every test image. If the model could find the target image in top 200 pertinent images, the success rate is 1, otherwise it is 0: see (4). Regarding every test image's success rate, I calculated Model's weighted success rate($SR\_W_{tot)}$ and success rate (SR_{tot}): see (5) and (6).

III. Experimental Results

I used 37 MOIBs to test DHam. I used the images that can be challenging for a CBIR's model: See Figure 6. I use here "element" instead of "object/symbol/sign/written text". One element is chosen from every target test image and a similar image to that is selected as query image to test the model's strength. Query element may vary in material, camera angle, situation, angle, view, color, background with target element. Moreover, it is notable that two elements may not precisely share the same shape.

I also did this test with the same 37 test MOIBs for BEiT -L and ResNet50 [16] CBIR models and compared the 3 model's results. Every query image is compared with all 5000, COCO VAL 2017 images. Total amount of features that are extracted by DHam from COCO VAL 2017 dataset that contains 5000 images, are 44891, around 9 features from every image. It means in average DHam has detected about 8 separate objects from every image. One of the features is the whole image itself: see ResNet, BEiT and DHam's Scores in Table I. Test results of BEiT, ResNet and DHam with 37 MOIBs Approves DHam has better performance over BEiT and RESNET with Score average of 0.73, 70% and 175% higher than BEiT and ResNet Score Mean and with Standard Deviation (SD) of 0.36 that is lower than BEiT and ResNet Score SD: see table I and Figure 5.

TABLE I
SCORE'S MEAN AND VARIANCE OF RESNET, DHAM AND BEIT

Row	Model's Name	Mean_TIS	Standard Deviation_TIS
1	DHam	0.734320557	0.364554727
2	BEiT	0.432229965	0.418201999
3	ResNet	0.267421603	0.383660562

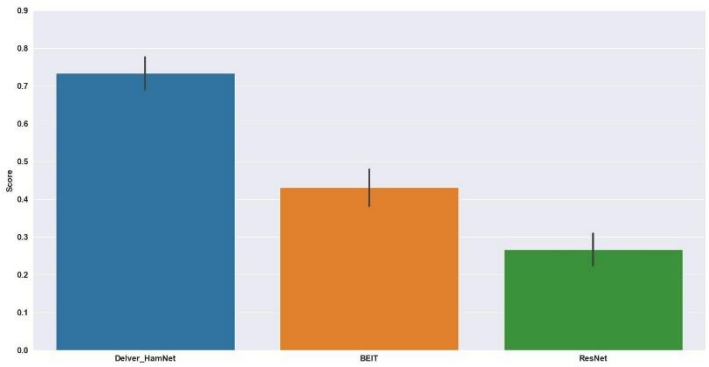

Fig. 5. DHam, BEiT and ResNet Score Bar Chart

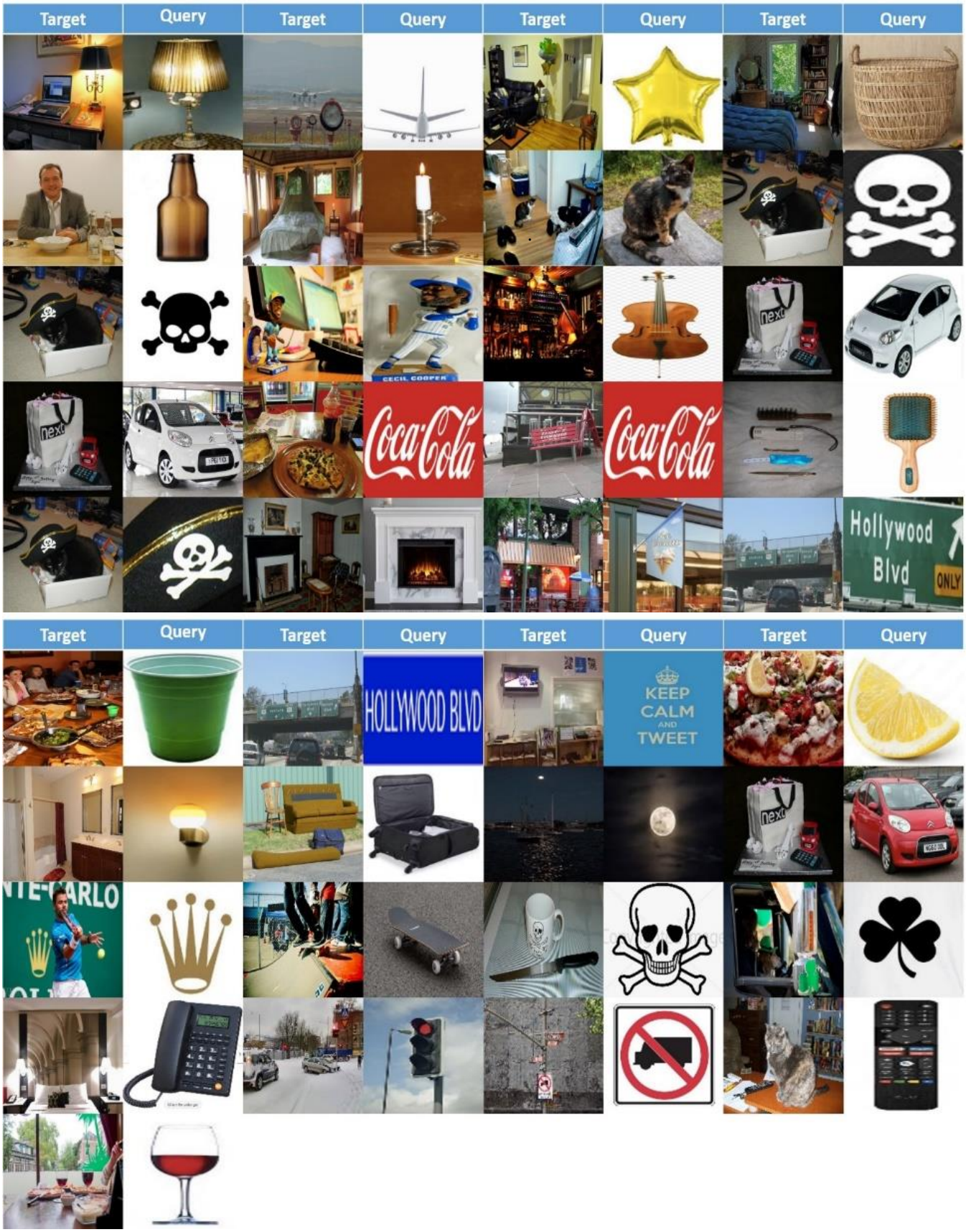

Fig. 6. 37 Test Query and Target Images**.**

TABLE II
RESNET, DHAM AND BEIT RESULTS

Row	Object/Symbol Name_COCO Filename	ResNet	Delver_HamNet	BEiT	Difficulty Level	ResNet_Score	Delver_Hamnet_Score	BEiT_Score
1	Abajour_16439	1	3	5	5	1	1	1
2	Airplane_33114	0	10	20	6	0	1	0.95
3	Ballon_182162	0	5	116	10	0	1	0.45
4	Basket_632	0	3	16	8	0	1	0.95
5	Blue HollyWood Blvd_1532	0	102	195	10	0	0.5	0.05
6	Bottle_50811	0	5	60	8	0	1	0.75
7	Candle_10092	0	67	0	10	0	0.7	0
8	Cat_189806	41	10	108	8	0.8	1	0.5
9	Cat_Skeleton_Sign_1_108244	0	51	0	9	0	0.75	0
10	Cat_Skeleton_Sign_108244	107	0	0	9	0.5	0	0
11	Cecil_Cooper_Toy_127987	0	1	0	8	0	1	0
12	Cello_117719	0	2	2	6	0	1	1
13	Citroen_1_180878	0	0	167	9	0	0	0.2
14	Citroen_180878	0	0	0	9	0	0	0
15	Coca_Cola_194724	0	25	12	8	0	0.9	0.95
16	Coca_Cola_453584	103	34	0	7	0.5	0.85	0
17	Comb_64574	33	1	2	7	0.85	1	1
18	Cropped_Cat_Skeleton_108244	42	1	1	3	0.8	1	1
19	Fireplace_16958	5	9	14	7	1	1	0.95
20	Flag_129062	10	19	11	9	1	0.95	0.95
21	Green HollyWood Blvd_1532	25	1	2	5	0.9	1	1
22	Green_Plastic_Glass_127394	0	6	0	7	0	1	0
23	Keep_Calm_121586	0	3	0	10	0	1	0
24	Lemon_159112	0	41	11	7	0	0.8	0.95
25	Light _6213	76	1	83	8	0.65	1	0.6
26	luggage_81061	37	60	0	10	0.85	0.75	0
27	Moon_79837	100	17	13	4	0.55	0.95	0.95
28	Red_Citroen_180878	0	149	0	9	0	0.3	0
29	Rolex_127530	44	0	73	6	0.8	0	0.65
30	Skate_257084	0	13	37	7	0	0.95	0.85
31	Skeleton_Symbol_2592	49	9	0	9	0.8	1	0
32	Suit of Clubs_Sign_53529	0	0	0	10	0	0	0
33	Telephone_231549	0	115	0	9	0	0.45	0
34	Traffic_Light_292997	0	132	72	7	0	0.35	0.65
35	Truck_Sign_544605	0	1	91	8	0	1	0.55
36	Tv_Remote_Control_520531	0	13	87	9	0	0.95	0.6
37	Wine_99039	182	5	20	6	0.1	1	0.95

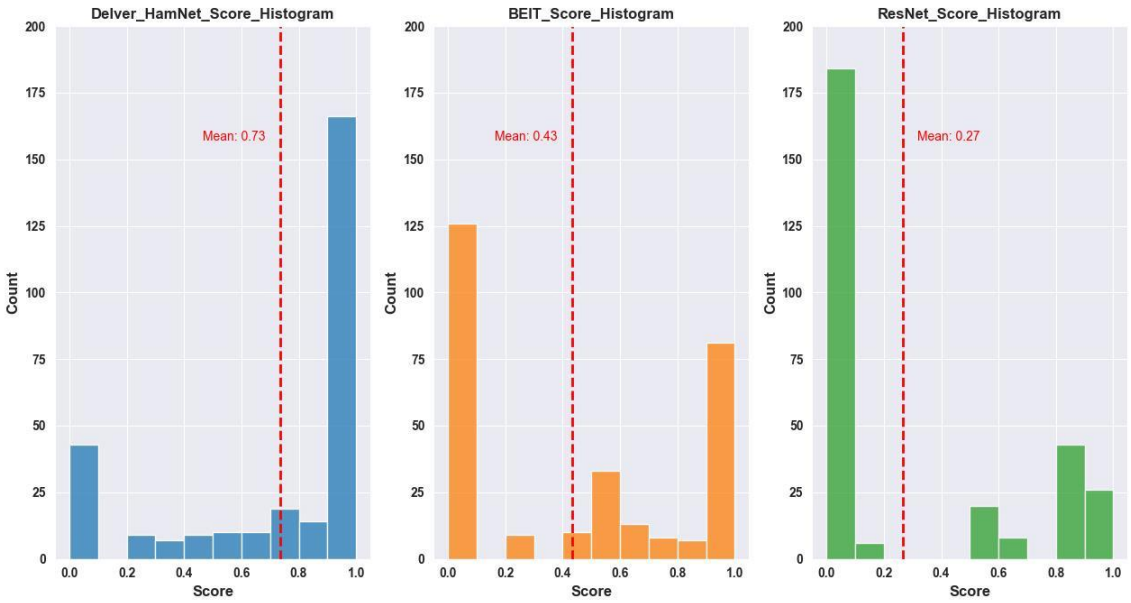

Fig. 7. DHam, BEIT and ResNet Histogram

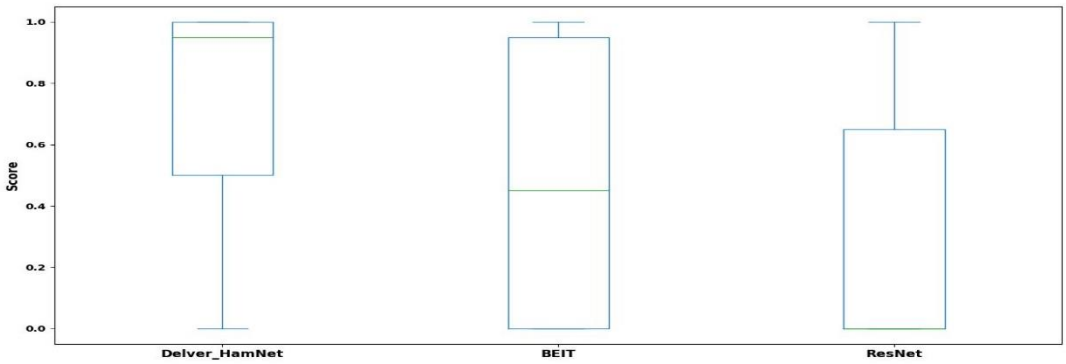

Fig. 8. DHam, BEIT and ResNet Boxplot

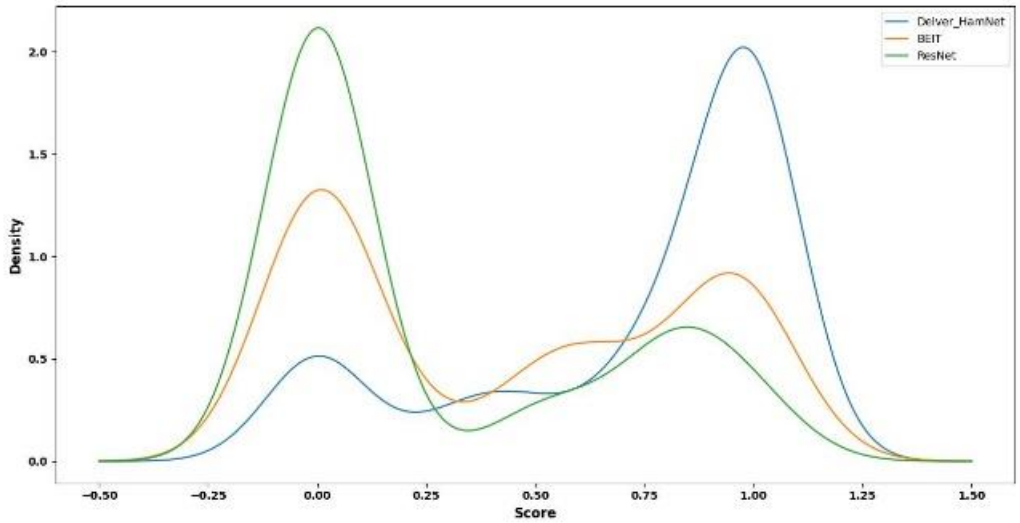

Fig. 9. DHam, BEIT and ResNet Density Plot

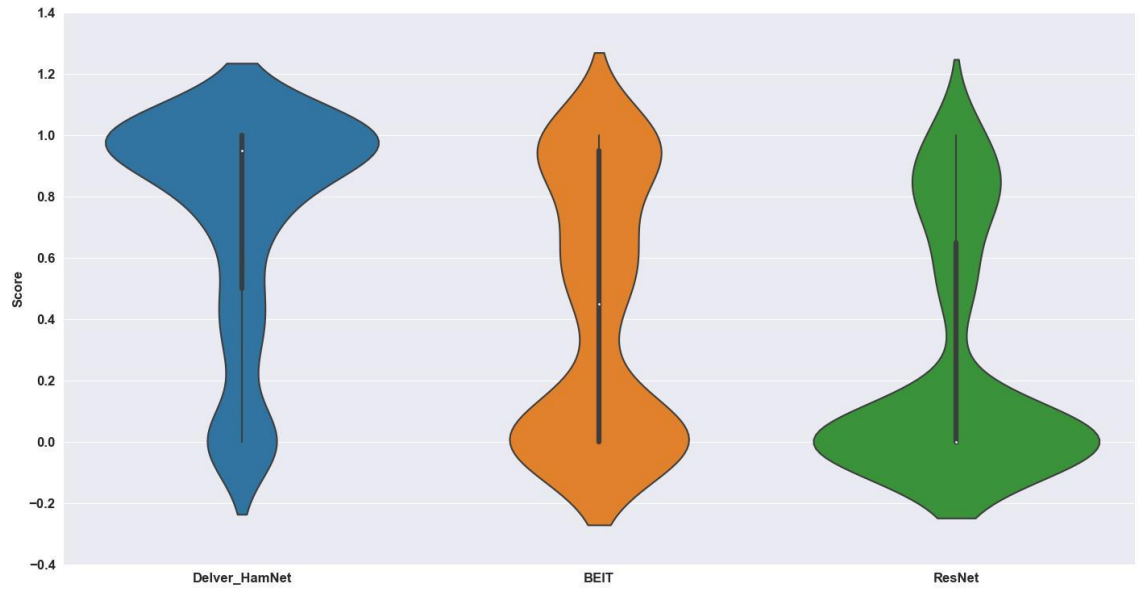

Fig. 10. DHam, BEiT and ResNet Violin Plot

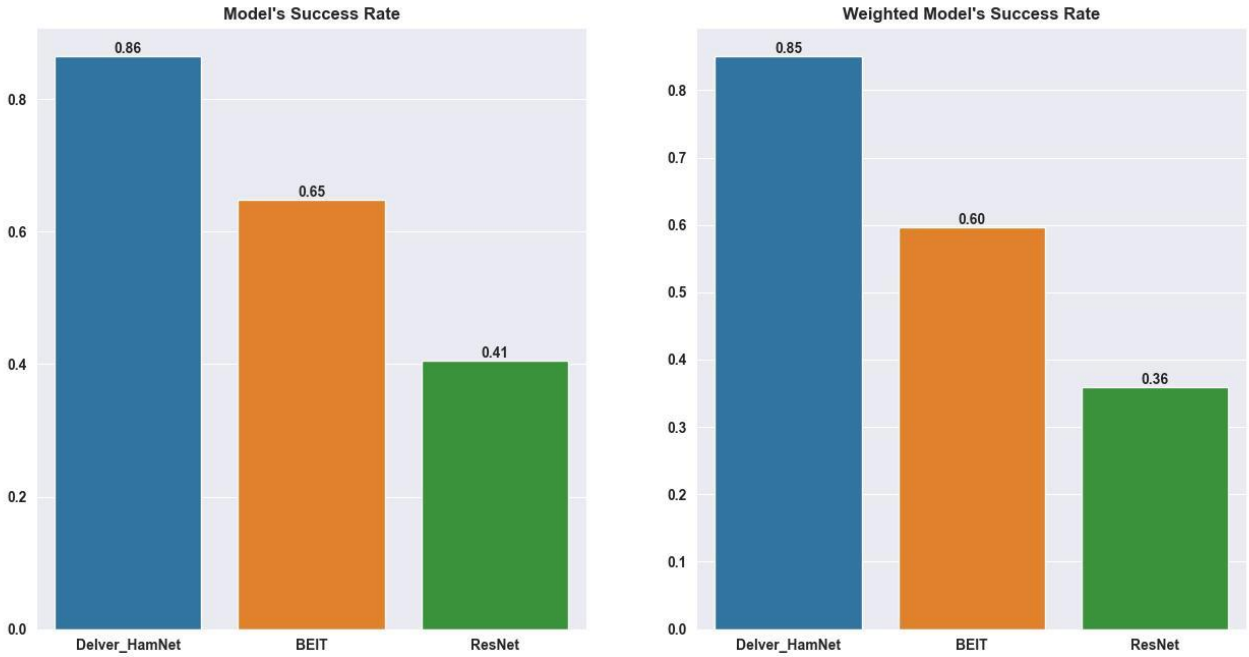

Fig. 11. DHam, BEiT and ResNet Success Rate Bar Chart

I calculated SR_W_{tot} and SR_{tot} for 3 models. In DHam congruence between SR_W_{tot} and SR_{tot} is conspicuous, approaching an approximate equivalence of 85%, as delineated in Figure 11. Accordingly, it is permissible to assert that the efficacy of DHam in discerning the target image reached a commendable success rate of 85% throughout this experimental evaluation.

IV. Computation Time

Since the computation time is important in a CBIR models, I measured the DHam's computation time. I also measured computation time with annoy [4]: see table III. Annoy reduced the computation time but reduced the test data score and success rate results reasonably. Therefore I used cosine distance for DHam.
The computation is done with an Intel Core i7 2nd generation, 8 Gig ram, no GPU laptop. Definitely the computation time can be reduced with parallel computing and big data but we can conclude that DHam is not a fast model.

. Conclusion

I introduce DHam, a joint model for CBIR. My model is simple to build and does not need any training data. DHam practically works. DHam is more prominent when you deal with MOIB's. One of The most significant characteristics of DHam is that it may bring irrelevant images but it rarely misses the target image. DHam pushes the state-of-the-art in CBIR. Test results showed that DHam performed better than pure BEiT and pure ResNet CBIR models .On the other hand, computation results prove that this is not a fast model. Besides, test results shows that DHam exhibits limitations when presented with non-object images. It encountered difficulties in detecting signs and written text imprinted on a surface. This is anticipated, given that my model is designed to identify analogous objects within an image rather than symbols, text, or signs.
In terms of future endeavors, the DHam holds promise for application in specialized domains through transfer learning. Furthermore, it can be further enhanced to accurately identify symbols, signs, or text printed on a surface in a MOIB. DHam has not undergone testing within a large-scale database, but such evaluation can be conducted in a sizable dataset. Moreover, efforts can be directed towards enhancing the model's processing speed.

Table III
DHam Computation time (Seconds)

Number of Features	TIME_1	TIME_2	TIME_3	TIME_4	TIME_5	mean	SD	Annoy_Average_Time
6,519	20	19	21	21	20	20.2	0.83666003	20
29,198	18	19	19	19	19	18.8	0.4472136	21
44,891	19	20	19	18	19	19	0.70710678	20
51,410	20	20	21	19	21	20.2	0.83666003	20
150,000	22	21	27	21	20	22.2	2.77488739	20
300,000	24	23	23	24	23	23.4	0.54772256	22
467,168	30	25	26	28	27	27.2	1.92353841	26
600,000	89	87	85	78	56	79	13.5092561	28
800,000	85	96	98	73	91	88.6	10.0647901	28
1,000,000	118	115	112	116	109	114	3.53553391	28
1,250,000	158	147	148	162	155	154	6.44204936	34
1,500,000	224	228	197	222	209	216	12.7867119	Ram_Limited(Failed)
1,868,672	269	287	261	275	273	273	9.48683298	Ram_Limited(Failed)

REFERENCES

[1] A. T. da Silva, A. X. Falcão, L. P. Magalhães, "Active learning paradigms for CBIR systems based on optimum-path forest classification," Pattern Recognition, Volume 44, Issue 12, Dec. 2011, doi: 10.1016/j.patcog.2011.04.026

[2] S. Zitan, I. Zeroual, S. Agoujil, "Performance investigation of a proposed CBIR search engine using deep convolutional neural networks," in Proc. CBI 2022: 7th International Conference on Business Intelligence, Khouribga, Morocco, May 26-28, 2022, pp. 41-49, doi.org/10.1007/978-3-031-06458-6_3

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is All You Need," in Proc of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 2017, pp. 5998–6008.

[4] S. Zitan, I. Zeroual, S. Agoujil. "A New CBIR Search Engine with a Vision Transformer Architecture". Artificial Intelligence and Smart Environment. ICAISE 2022. Lecture Notes in Networks and Systems, Volume 635, pp. 64-69. March, 2023. Springer, Cham. doi:10.1007/978-3-031-26254-8_9

[5] D. Srivastava, S. S. Singh, B. Rajitha, M. Verma, M. Kaur and H. -N. Lee, "Content-Based Image Retrieval: A Survey on Local and Global Features Selection, Extraction, Representation, and Evaluation Parameters," in *IEEE Access*, vol. 11, pp. 95410-95431, July 2023, doi: 10.1109/ACCESS.2023.3308911

[6] A. El-Nouby, N. Neverova, I. Laptev, H. Jégou, "Training Vision Transformers for Image Retrieval," 2021, arXiv:2102.05644.

[7] Gkelios, Socratis et al. "Investigating the Vision Transformer Model for Image Retrieval Tasks." *2021 17th International Conference on Distributed Computing in Sensor Systems (DCOSS)* (2021): 367-373.

[8] E. S.Sabry, S. S.Elagooz et al. "Image Retrieval Using Convolutional Autoencoder, InfoGAN, and Vision Transformer Unsupervised Models." *IEEE Access* 11 (2023): 20445-20477.

[9] S. Fadaei et al. "A New Content-Based Image Retrieval System Based on Optimized Integration of DCD, Wavelet and Curvelet Features." IET Image processing, Volume 11, Issue 2, (2017):89-98.

[10] U. Salahuddin, W. Xingyuan, W. Chunpeng & W. Yu, "A Decisive Content Based Image Retrieval Approach for Feature Fusion in Visual and Textual Images," Knowledge-Based Systems, 179, May 2019, doi:10.1016/j.knosys.2019.05.001.

[11] B. C. Mohan, T. K. Chaitanya and T. Tirupal, "Fast and Accurate Content Based Image Classification and Retrieval using Gaussian Hermite Moments applied to COIL 20 and COIL 100," *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Kanpur, India, 2019, pp. 1-5, doi: 10.1109/ICCCNT45670.2019.8944775.

[12] S. Fadaei, A. Rashno and E. Rashno, "Content-based Image Retrieval Speedup," *2019 5th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS)*, Shahrood, Iran, 2019, pp. 1-5, doi: 10.1109/ICSPIS48872.2019.9066132.

[13] X. Li, J. Yang, J. Ma, "Recent developments of content-based image retrieval (CBIR)," Neurocomputing, vol. 452, pp.675-689, Sep. 2021, doi: 10.1016/j.neucom.2020.07.139.

[14] X. Zhou, R. Girdhar, A. Joulin, P. Krahenbuhl, "Detecting Twenty-Thousand Classes Using Image-Level Supervision" in Computer Vision" in Proc. Computer Vision –ECCV, Tel Aviv, Israel, 2022, pp. 350-368

[15] H. Bao, L. Dong, S. Piao, F. Wei, "Beit: Bert pre-training of image transformers" Presented at the 10th Int. Conf. Learning Representations, Conference paper 2678, [online], Apr. 25-29, 2022

[16] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

[17] Tsung-Yi Lin, et al, 2017, "COCO 2017: Common Objects in Context 2017", Microsoft, https://cocodataset.org/#home

Mostafa Jelveh was born In Tehran, Iran. He received the B.Sc. degree (Hons.) in industrial engineering (Industrial Production) from the Faculty of Industrial Engineering, K. N. Toosi University of Technology, Tehran, Iran, in 2006, the M.Sc. degree in industrial-industrial engineering from the Faculty of Engineering, Shahed University, Tehran, Iran, in 2009. He is currently working as a Data Scientist and Intellectual Property Examiner at Iran Intellectual Property Center, Tehran, Iran. He has competence utilizing python and R to create machine learning, deep learning, RNN (Recurrent Neural Network) and Transformers based models across a variety of phases and applications. His research interests include, artificial intelligence, deep learning, RNN, Transformers, NLP (Natural Language Processing) and LMMs (Large Multimodal Models).