JOIV : Int. J. Inform. Visualization, 7(1) - March 2023 115-121

INTERNATIONAL JOURNAL
ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv

Batik Images Retrieval Using Pre-trained model and K-Nearest Neighbor

Agus Eko Minarno [a], Muhammad Yusril Hasanuddin [a], Yufis Azhar [a,*]

[a] *Universitas Muhammadiyah Malang, Malang, Indonesia*

*Corresponding author: *yufis@umm.ac.id*

***Abstract*— Batik is an Indonesian cultural heritage that should be preserved. Over time, many batik motifs have sprung up, which can lead to mutual claims between craftsmen. Therefore, it is necessary to create a system to measure the similarity of a batik motif. This research is focused on making Content-Based Image Retrieval (CBIR) on batik images. The dataset used in this research is big data Batik images. The authors used transfer learning on several pre-trained models and used Convolutional Neural Network (CNN) Autoencoder from previous studies to extract features on all images in the database. The extracted features calculate the Euclidean distance between the query and all images in the database to retrieve images. The image closest to the query will be retrieved according to the number of r, namely 3, 5, 10, or 15. Before the image is retrieved, the retrieval system is used to re-ranked with K-Nearest Neighbor (KNN), which classifies the retrieved image. The results of this study prove that MobileNetV2 + KNN is the best model in terms of Image Retrieval Batik, followed by InceptionV3 and VGG19 as the second and third ranks. Moreover, CNN Autoencoder from previous research and InceptionResNetV2 are ranked fourth and fifth. In this study, it was also found that the use of KNN re-ranking can increase the precision value by 0.00272. For further research, deploying these models, especially for MobileNetV2 is an approach for seeing a major impact on batik craftsmanship for decreasing batik motif plagiarism.**

***Keywords*—Batik; content based image retrieval; autoencoder; KNN; CNN.**

Manuscript received 18 Oct. 2022; revised 14 Jan. 2023; accepted 22 Jan. 2023. Date of publication 31 Mar. 2023.

I. Introduction

Batik is a traditional art form that has been passed down for generations in Indonesia. It is a method of creating intricate patterns and designs on fabric using wax and dye. The process of batik-making is complex and time-consuming, but the results are truly beautiful and unique. The history of batik can be traced back to ancient civilizations in Indonesia, where it was used to decorate clothing, textiles, and other items. The word "batik" comes from the Javanese word "*mbatik*," which means "to write" or "to dot." This refers to the process of applying wax to the fabric using a "canting," a small metal or bamboo tool with a spout that is used to draw the wax onto the fabric. Once the wax is applied, the fabric is dipped into dye, which resists the dye, creating the desired pattern. This process is repeated multiple times with different colors of dye to create intricate and detailed designs. The result is a piece of fabric that is rich with color and patterns and truly unique.

Batik is known for its vibrant colors and unique designs and is often used to make clothing, home decor, and other items. The art of batik is recognized as a UNESCO Intangible Cultural Heritage and is an important part of Indonesian culture. It is considered a traditional art that is still relevant and useful in the modern era and has been adopted by many countries as their own traditional art [1].

The traditional method of batik has been passed down from generation to generation, and today many talented batik artists in Indonesia continue to create beautiful and unique designs. Some of the famous batik motifs are from regions such as Java, Sumatra, Bali, and Sulawesi, and each region has its characteristic motifs, colors, and techniques.

Indonesia has much cultural heritage. There are many types of batik found in Indonesia, and there are at least 15 variations of batik found in the province of Java [2]. There are also various methods in batik, such as the stamp or writing method. The many variations of patterns in batik can lead to mutually claiming motives between craftsmen [3] [4]. Therefore, we need a system that can indicate whether a batik motif has been claimed or not.

The development of information technology, in this case, Artificial Intelligence, can overcome this problem. One of the artificial intelligence techniques that can help with this problem is Image Retrieval. Before the development of image retrieval technology, one of the methods used to search a collection of images using image search was text-based. This technique is called image caption information retrieval. This technique will return the image based on the caption [5] [6]. Like image caption information retrieval, image retrieval is a process of finding images from a database based on image input queries [7]. The purpose of image retrieval is the same as image caption information retrieval, and the difference is that image retrieval uses a query in the form of an image, while image caption information retrieval uses a text query.

The image return the authors used in this paper is CBIR (Content Based Image Retrieval) which searches for images based on the components that make up the image. Components in CBIR consist of color, shape, texture, topology, etc. [8]. CBIR is also an image search technique that does not rely on manually defined annotations [9]. Therefore, image returns can be classified as supervised or unsupervised learning. The difference between supervised learning and unsupervised learning is in determining labels on the dataset. If the dataset has a label, then it is included in supervised learning. On the other hand, unsupervised learning aims to group data in unlabeled datasets [10], [11]. In this paper, the authors used unsupervised learning on image recovery.

The method the authors used to retrieve the image of batik is transfer learning on CNN. Transfer learning is a CNN model architecture that has been tested on many datasets and implemented on new datasets [12]–[17]. Previously, there had been CBIR for batik. Prasetyo and Kardihasin [18] used CNN and obtained an Average Precision Recall (APR) of 0.7654. The dataset used in this study is a personal dataset with a total of 1552 images consisting of 97 classes. Each class has 16 pictures. In addition, there is also research by Azhar et al. that uses the Multi Texton Co-occurrence Descriptor (MTCD) method with KNN to retrieve images on the Batik300 dataset. The Batik300 dataset consists of 50 classes, with each class having six images. This study proves that this method increases the precision value by 0.8% [19].

From the two references, the dataset used is still small, and adding the number of datasets is needed to increase the variation of the data. Therefore, the authors propose using transfer learning and KNN in CBIR with the Batik7K dataset. IR enables it to retrieve images effectively and efficiently from large-scale databases with input images [20]. This study analyzes the effectiveness of the transfer learning method in CBIR with a large number of datasets. In this study, the authors open opportunities for another researcher to use the Batik7K dataset.

II. Material and Method

A. Data Gathering

The dataset used in this paper is the Batik7K dataset. The Batik7K dataset was obtained from the filtering dataset Batik41K [21]. The number of classes in the Batik7K dataset is still the same as Batik41K, which is 355 classes. The difference is that the number of images per class on Batik7K is 20.

Therefore, the total number of images used in this paper is 7100. Furthermore, this dataset is divided into training data and test data, and the division is 80:20. Thus, the number of training images per class is 16, while the test data is 4. The dimensions of each image in the dataset are 600 x 400. The sample of the dataset can be seen in Fig 1.

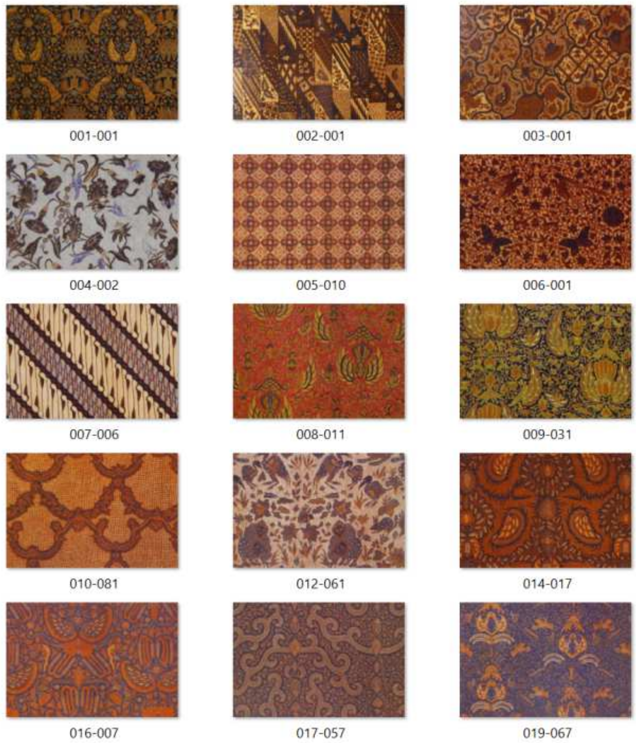

Fig. 1 Sample dataset Batik 7K

B. Data Processing

Data processing includes data collection, resizing data, and normalizing data. Data collection is done using the skiimage.io library, which converts the pixels in the image into an array form. After that, the data will enter the resizing process. At this stage, the array will be resized to 128 x 128. This resizing process is carried out with the help of the *skiimage* transform library. Furthermore, the *ndarray* value goes through the normalization process on the color features. Therefore, the array value is divided by 255. So, the array value in each image will be in the form of a 128 x 128 matrix with normalized color values. From Fig 2, the program to process data can be seen.

```
Program Processing Data per Image
image = skimage.io.imread(filePath, as_gray=False)
image = skimage.transform.resize(image, (128,128))
image = image / 255
```

Fig. 2 Program Processing Data per Image

C. Feature Extraction

After going through the data processing process, the image that has become an array form is carried out by the extraction process on its features. The extraction process aims for the system to gain knowledge so that the system has good knowledge of class varieties in the batik dataset. The process of feature extraction starts by converting the array into a *NumPy* array (*ndarray*) according to the input values in the model. The input value in the model is the same as the resize value. Next, the *NumPy* array will be extracted according to the transfer model used (the transfer learning models we used are discussed in section 2.4). Later obtained extraction according to the output layer on transfer learning. The result of the next output layer is flattened. The purpose of flattening is to convert the extraction output into a one-dimensional matrix.

D. Models

We used transfer learning from the pre-trained model used to extract features in the image. In this paper, the authors tested five transfer learning models and one CNN AutoEncoder model, as made by Prasetyo and Akardihas. [18]. We used the study's models of MobileNetV2, InceptionV3, InceptionResNetV2, and VGG19. As has been studied by Bose et at., these five pre-trained models have a good ability to extract features [22].

1) MobileNetV2: MobileNetV2 was introduced by Putra et al. [23]. The structure of MobileNetV2 can be seen in Fig 3. MobileNetV2 has been tested on ImageNet classification. In addition, MobileNetV2 was also tested on pattern recognition of traditional clothes [24] and batik image classification [25]. The MobileNetV2 model uses a convolution block with a unique property that separates the model network's expressiveness capacity by using an input bottleneck [26].

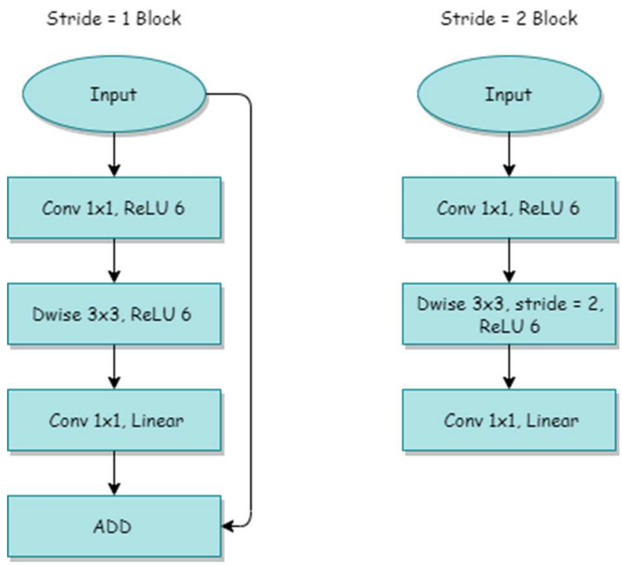

Fig. 3 Model structure MobileNetV2 [26]

2) InceptionV3 and InceptionResNetV2: InceptionV3 was introduced by Widyantoko et al. [25]. InceptionV3 enhances the model network by making the most efficient use of incremental computing with suitable factored convolution and aggressive regularization. InceptionV3 is suitable for big-data classification scenarios [25]. On the other hand, InceptionResNetV2 was introduced by Widyantoko et al. [25]. InceptionResNetV2 is an Inception model added with a residual layer, and ts is aimed at the computational load on the Inception model [25].

3) VGG19: VGG19 was introduced by Arsa and Susila [27]. VGG19 can generalize well to various tasks and data sets, matching or outperforming more complex recognition channels built around image representation. The results of the VGG19 model emphasize the depth of visual representation [27].

4) CAE: In Table 1, the structure of the CAE model, as found in Prasetyo and Kardihasin [18], is a CNN layer that contains an encoder layer and a decoder layer. In this method, because Prasetyo and Kardihasin's [18] paper does not include the number of epochs, the authors used 100 epochs for training data.

TABLE I
CAE [18]

Layer Type	Size	Output Shape
Input	(128,128,3)	-
Convolutional + Relu	32 (3x3) filters, 1 stride, 2 padding	(128,128,32)
Max Pooling + Dropout	32 (2x2) filters, 2 stride, 0 padding	(64,64,32)
Convolutional + Relu	64 (3x3) filters, 1 stride, 2 padding	(64,64,64)
Max Pooling + Dropout	64 (2x2) filters, 2 stride, 0 padding	(32,32,64)
Convolutional + Relu	64 (3x3) filters, 1 stride, 2 padding	(32,32,64)
Max Pooling + Dropout	64 (2x2) filters, 2 stride, 0 padding	(16,16,64)
Convolutional + Relu	128 (3x3) filters, 1 stride, 2 padding	(16,16,128)
Max Pooling + Dropout	128 (2x2) filters, 2 stride, 0 padding	(8,8,128)
Max Pooling (Neural Code)	128 (2x2) filters, 1 stride, 2 padding	(4,4,128)
Unpoling	128 (2x2) filters, 2 stride, 0 padding	(8,8,128)
Deconvovution + Relu	128 (3x3) filters, 1 stride, 2 padding	(8,8,128)
Unpooling + Dropout	128 (2x2) filters, 2 stride, 0 padding	(16,16,128)
Deconvovution + Relu	64 (3x3) filters, 1 stride, 2 padding	(16,16,64)
Unpooling + Dropout	64 (2x2) filters, 2 stride, 0 padding	(32,32,64)
Deconvovution + Relu	64 (3x3) filters, 1 stride, 2 padding	(32,32,64)
Unpooling + Dropout	64 (2x2) filters, 2 stride, 0 padding	(64,64,64)
Deconvovution + Relu	32 (3x3) filters, 1 stride, 2 padding	(64,64,32)
Unpooling + Dropout	32 (2x2) filters, 2 stride, 0 padding	(128,128,32)
Deconvovution + Sigmoid	32 (3x3) filters, 1 stride, 2 padding	(128,128,3)

E. Similarity Measure

Image similarity measurements were calculated using Euclidean distances [28]. The image is returned by calculating the Euclidean distance from the query with other images. The images with the smallest Euclidean distance value are the most similar to the query. The measurement of the Euclidean distance value is calculated by calculating the difference in the feature extraction results in each image. Therefore, the query must perform a feature extraction process in its implementation. Euclidean Distance gives the distance between two points directly with a straight line [29]. Euclidean Distance formula can be seen in formula (1).

$$d(P,Q)=\|P-Q\|_0=\sqrt{\sum_{i=1}^{n}(p_i-q_i)^2} \quad (1)$$

Where, d is the Euclidean distance from the point P to Q and n is the dimension of the vector while p and q are the numerical points of the dimension n.

F. K-Nearest Neighbor (KNN)

K-Nearest Neighbors (KNN) was developed from the need to perform discriminant analysis when reliable parametric predictions of probability density are unknown or difficult to determine. K-Nearest Neighbors (KNN) is a technique for classifying data that previously did not know the distribution of the data. The use of KNN in this paper is intended to adjust the return position, so that good returns are obtained. Azhar et al. [19] prove that KNN can increase precision by re-ranking the retrieved image.

G. Evaluation

To show whether the model that the authors built is a good model or not. The authors used the Precision test [30] the precision formula is shown in formula (2).

$$\text{Precision}=\frac{TP}{TP+FP} \tag{2}$$

Where TP and TN are True Positive, and True Negative in both formulas.

Precision is used to determine the ratio of the number of relevant images to the total number of images taken [31]. This ratio is the basis for measuring how good the model is made. The greater the Precision value, the more it indicates the model being tested is good.

III. Results and Discussion

A. Experiment Settings

The authors implemented the method on a computer device with 32 GB RAM, Intel® Core™ i9-10900K processor. The method is implemented in Python programming language with *jupyter* notebook software using GPU as a runtime device. While the library used is *Keras Tensorflow*. The type of GPU used is NVIDIA GeForce RTX 2060.

B. Method Implementation

The experiment starts with entering training data and test data into image processing, after which the features are extracted. Furthermore, the Euclidean distance measurement to the query image is carried out with all the data in this dataset, where a query is the entire unit of test data.

Contribution to this research is the application of the transfer learning method with KNN on Big Data Batik. The value of precision on image return depends on the value of K, where K is the number of k-neighbors of image return. In this study, precision is used to measure whether the method that the authors implemented impacts the image return performance. Recall is not used as a benchmark in this study because if the recall is used, then the image returns must be by the maximum number per class. In this study, the authors used retrieval values ranging from small to large to determine the effect of the K value on the development of the precision value.

The experiment was done by trying research methods with several scenarios. The authors used two scenarios to prove whether KNN can improve precision performance. Thus, two scenarios are tested for each transfer learning model and one CAE model. In each scenario, there are variations in the number of images returned (r) to determine how well the model is in the case of small and large images.

TABLE II
PRECISION ON EACH MODEL WITHOUT USING RE-RANKING KNN

r	Precision				
	CAE	MobileNetV2	InceptionV3	InceptionResNetV2	VGG19
3	0.9387	0.9690	0.9605	0.9309	0.9535
5	0.9492	0.9725	0.9675	0.9380	0.9576
10	0.9528	0.9774	0.9766	0.9436	0.9598
15	0.9598	0.9802	0.9788	0.9471	0.9626

In the first scenario, all models were tested without the KNN re-ranking method. Here, r is the number of images returned from the input query. It can be seen in Table 2 that MobileNetV2 has the highest precision compared to other methods.

TABLE III
PRECISION ON EACH MODEL USING RE-RANKING KNN

r	K	Precision				
		CAE + KNN	**MobileNetV2 + KNN**	**InceptionV3 + KNN**	**InceptionResNetV2 + KNN**	**VGG19 + KNN**
3	3	0.9387	0.9690	0.9605	0.9309	0.9535
	5	0.9443	0.9725	0.9676	0.9379	0.9577
	10	0.9514	0.9761	0.9767	0.9436	0.9598
	15	0.9605	0.9809	0.9767	0.9436	0.9626
5	5	0.9492	0.9725	0.9675	0.9380	0.9576
	10	0.9492	0.9760	0.9768	0.9437	0.9598
	15	0.9591	0.9809	0.9795	0.9472	0.9627
10	10	0.9528	0.9774	0.9766	0.9436	0.9598
	15	0.9584	0.9809	0.9796	0.9471	0.9626
15	15	0.9598	0.9802	0.9788	0.9471	0.9626
	20	0.9612	0.9830	0.9802	0.9542	0.9633

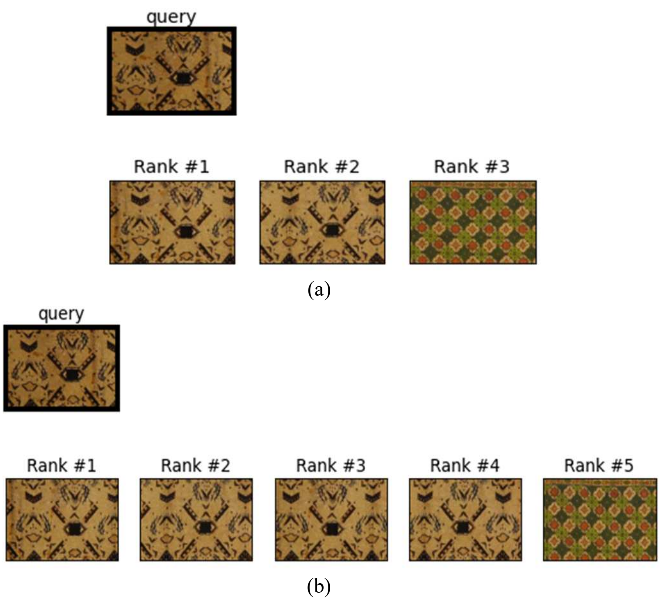

Fig. 4 Comparison of Image Retrieval. (a) with re-ranking KNN; (b) without re-ranking KNN

In Table 3, K is the number of KNN neighbors, and r is the number of images returned. The batik image retrieval system first takes a number of pictures with a K value, then KNN will classify each image taken and sort them according to the classified label. After that, the system returns the image according to the number r. Fig 4 is an example of Batik image retrieval in two conditions, namely by using KNN re-ranking and without using KNN re-ranking with the same number of returns, namely three images.

In Fig 4 (a), it can be seen that the third-ranked image returned does not match the image entered in the query. On the other hand, in Fig 4 (b), which uses KNN re-ranking with a total of 5 k-neighbors, KNN functions to classify images taken according to the k-neighbor value. From this classification process, the returned images are re-ranked before being returned according to the number of r as much as 3.

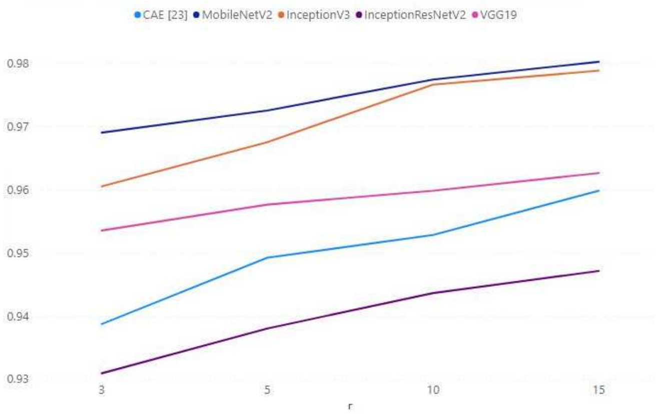

Fig. 5 Precision of each model based on the value of r using train and test data 80:20

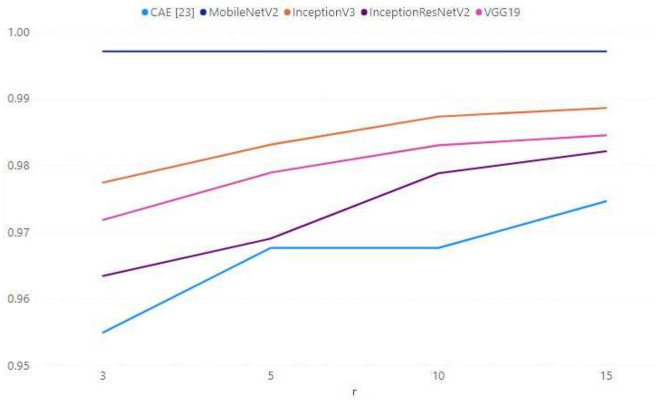

Fig. 6 Precision of each model based on the value of r using train and test data 90:10

As shown in Fig 5 and 6, MobileNetV2 got the first rank in terms of Image Retrieval Batik without using Re-Rank KNN from all variations of the r value. The next ranking was followed by InceptionV3, VGG19, CAE, and InceptionResNetV2 in second, third, fourth, and fifth positions, respectively.

As can be seen in Fig 7 and 8, the order in which the precision values of the model and KNN are ranked is still the same as the order of the precision values of the model without KNN. The first position is held by MobileNetV2 + KNN, followed by InceptionV3 + KNN, VGG19 + KNN, CAE + KNN, and InceptionResNetV2 + KNN in the second, third, and fourth positions, respectively, taken from the largest precision value that can be generated for each variation of r, from the graph, in the variation of r = 3 and r = 5, the precision value of InceptionV3 for K = 10, managed to outperform MobileNetV2 for the same K value of K = 10.

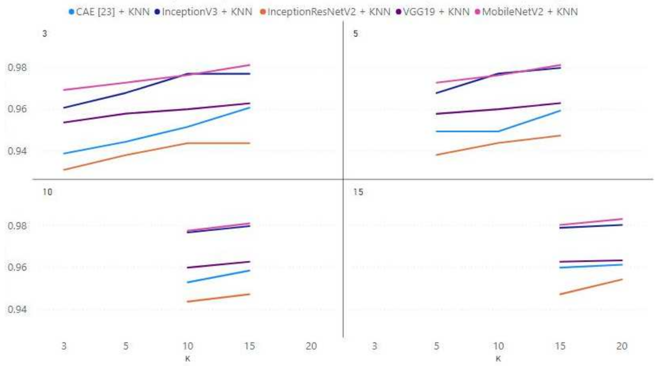

Fig. 7 Precision of all models based on the values of r and K using train and test data 80:20

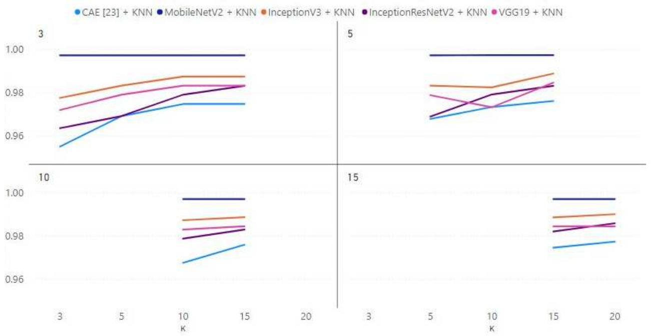

Fig. 8 Precision of all models based on the values of r and K using train and test data 80:20

To find out how influential KNN has on precision values in all tested models, the difference in precision values for models using KNN and not using KNN is calculated and averaged from the two splitting scenarios. The results can be seen in Table 4. As a result, KNN re-ranking can increase the precision model value by 0.00272 compared to models that do not use KNN re-ranking. Furthermore, the average increase in precision values between those using and not using the KNN re-rank for all pre-trained models tested was 0.00237. And specifically for the CAE model, there was an increase of 0.0041 between those using KNN re-ranking and not using KNN re-ranking.

TABLE IV
AVERAGE PRECISION VALUES FOR ALL R AND K

Model	With re-ranking KNN	Without re-ranking KNN	Difference between precision values with and without KNN re-rank
CAE	0.9622	0.9581	0.0041
MobileNetV2	0.9871	0.9859	0.0012
InceptionV3	0.9802	0.9774	0.0028
InceptionResNetV2	0.9605	0.9566	0.0039
VGG19	0.9705	0.9689	0.0016
Average Difference			0.00272

Fig 9. Retrieval result using MobileNetV2

C. *Further Analysis*

The authors provide a collection of images returned using the MobileNetV2 and CAE methods in Fig 9 and Fig 10. Judging from the comparison of Fig 9 and Fig 10, the Image Retrieval Batik using MobileNetV2 is better than the method in the previous study, namely CAE. There is a significant difference between the results of the MobileNetV2 model and the CAE model. In CAE, the images returned at positions 7, 8, 9, and 10 do not match the images entered by the query. In contrast to MobileNetV2, which returned the ten images correctly according to the input query.

One of the key advantages of MobileNetV2 over CAE is its ability to perform well on a wide range of tasks. MobileNetV2 has been trained on a large dataset and fine-tuned for various tasks, such as object detection, semantic segmentation, and image classification. This allows it to be used in a wide range of applications, from mobile applications to embedded systems. MobileNetV2 has been specifically designed to be efficient in terms of model size and inference time.

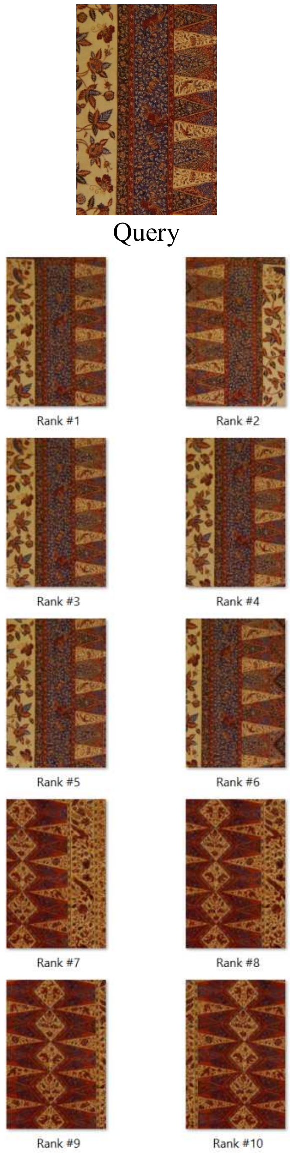

Fig 10. Retrieval result using CAE

IV. Conclusion

In implementing several pre-trained models and models contained in the main reference, MobileNetV2 is the best model in terms of Batik Image Retrieval, followed by InceptionV3 and VGG19 in the second and third positions. Meanwhile, CAE and InceptionResNetV2 each took fourth and fifth positions. This study also found that using KNN in Image Retrieval on the Batik 7K dataset can increase the precision value by 0.00272. For further research, there could be the approach for different pre-trained models and model deployment, especially CBIR using MobileNetV2 and KNN.

Acknowledgment

The authors thank Laboratory Informatics, University of Muhammadiyah Malang, for supporting this research.

References

[1] Director-General of UNESCO, "Indonesia Batik Certificate," *4. COM*, 2009. http://kwriu.kemdikbud.go.id/old_site/gambar/gb-info/sertifikat/ICH-03-Batik.png (accessed Jan. 19, 2022).

[2] A. E. Minarno, M. C. Mustaqim, Y. Azhar, W. A. Kusuma, and Y. Munarko, "Deep Convolutional Generative Adversarial Network Application in Batik Pattern Generator," in *2021 9th International Conference on Information and Communication Technology, ICoICT 2021*, 2021, pp. 54–59. doi: 10.1109/ICoICT52021.2021.9527514.

[3] A. E. Minarno, M. C. Mustaqim, Y. Azhar, W. A. Kusuma, and Y. Munarko, "Deep Convolutional Generative Adversarial Network Application in Batik Pattern Generator," *2021 9th International Conference on Information and Communication Technology, ICoICT 2021*, pp. 54–59, Aug. 2021, doi: 10.1109/ICOICT52021.2021.9527514.

[4] A. E. Minarno, F. D. S. Sumadi, H. Wibowo, and Y. Munarko, "Classification of batik patterns using K-nearest neighbor and support vector machine," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 3, pp. 1260–1267, 2020, doi: 10.11591/eei.v9i3.1971.

[5] P. Kaur, H. S. Pannu, and A. K. Malhi, "Comparative analysis on cross-modal information retrieval: A review," *Comput Sci Rev*, vol. 39, p. 100336, 2021, doi: 10.1016/j.cosrev.2020.100336.

[6] Y. Zhang, X. Wang, Z. Guo, and J. Li, "ImageSem at ImageCLEF 2018 caption task: Image retrieval and transfer learning," *CEUR Workshop Proc*, vol. 2125, 2018.

[7] M. Tzelepi and A. Tefas, "Deep convolutional learning for Content Based Image Retrieval," *Neurocomputing*, vol. 275, pp. 2467–2478, 2018, doi: 10.1016/j.neucom.2017.11.022.

[8] N. K. Rout, M. Atulkar, and M. K. Ahirwal, "A review on content-based image retrieval system: Present trends and future challenges," *Int J Comput Vis Robot*, vol. 11, no. 5, pp. 461–485, 2021, doi: 10.1504/IJCVR.2021.117578.

[9] M. S. Haji, M. H. Alkawaz, A. Rehman, and T. Saba, "Content-based image retrieval: A deep look at features prospectus," *Int J Comput Vis Robot*, vol. 9, no. 1, pp. 14–38, 2019, doi: 10.1504/IJCVR.2019.098004.

[10] J. Delua, "Supervised vs. Unsupervised Learning: What's the Difference?," *IBM*, 2021. https://www.ibm.com/cloud/blog/supervised-vs-unsupervised-learning (accessed Jan. 20, 2022).

[11] A. E. Minarno, K. M. Ghufron, T. S. Sabrila, L. Husniah, and F. D. S. Sumadi, "CNN Based Autoencoder Application in Breast Cancer Image Retrieval," in *Proceedings - 2021 International Seminar on Intelligent Technology and Its Application: Intelligent Systems for the New Normal Era, ISITIA 2021*, 2021, pp. 29–34. doi: 10.1109/ISITIA52817.2021.9502205.

[12] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A Comprehensive Survey on Transfer Learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2021, doi: 10.1109/JPROC.2020.3004555.

[13] A. E. Minarno, I. S. Kantomo, F. D. S. Sumadi, H. A. Nugroho, and Z. Ibrahim, "Classification of Brain Tumors on MRI Images Using DenseNet and Support Vector Machine," *JOIV : International Journal on Informatics Visualization*, vol. 6, no. 2, pp. 404–410, Jun. 2022, doi: 10.30630/JOIV.6.2.991.

[14] A. E. Minarno, M. H. C. Mandiri, Y. Azhar, F. Bimantoro, H. A. Nugroho, and Z. Ibrahim, "Classification of Diabetic Retinopathy Disease Using Convolutional Neural Network," *JOIV : International Journal on Informatics Visualization*, vol. 6, no. 1, pp. 12–18, Mar. 2022, doi: 10.30630/JOIV.6.1.857.

[15] A. E. Minarno, B. Y. Sasongko, Y. Munarko, H. A. Nugroho, and Z. Ibrahim, "Convolutional Neural Network featuring VGG-16 Model for Glioma Classification," *JOIV : International Journal on Informatics Visualization*, vol. 6, no. 3, pp. 660–666, Sep. 2022, doi: 10.30630/JOIV.6.3.1230.

[16] A. E. Minarno, M. Rifal Alfarizy, A. Hendryawan, S. Syaifuddin, and Y. Munarko, "Pneumonia Classification using Gabor-Convolutional Neural Networks and Image Enhancement," *2021 9th International Conference on Information and Communication Technology, ICoICT 2021*, pp. 180–185, Aug. 2021, doi: 10.1109/ICOICT52021.2021.9527427.

[17] Y. Munarko and A. E. Minarno, "HII: Histogram inverted index for fast images retrieval," *International Journal of Electrical and Computer Engineering*, vol. 8, no. 5, pp. 3140–3148, 2018, doi: 10.11591/ijece.v8i5.pp.3140-3148.

[18] H. Prasetyo and B. A. Putra Akardihas, "Batik image retrieval using convolutional neural network," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 17, no. 6, pp. 3010–3018, 2019, doi: 10.12928/TELKOMNIKA.v17i6.12701.

[19] Y. Azhar, A. Eko Minarno, and Y. Munarko, "Re-Ranking Image Retrieval on Multi Texton Co-Occurrence Descriptor Using K-Nearest Neighbor," *Proceeding of the Electrical Engineering Computer Science and Informatics*, vol. 5, no. 5, pp. 589–593, 2018, doi: 10.11591/eecsi.v5i5.1683.

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

[21] A. E. Minarno, Y. Azhar, F. D. Setiawan Sumadi, and Y. Munarko, "A Robust Batik Image Classification using Multi Texton Co-Occurrence Descriptor and Support Vector Machine," *2020 3rd International Conference on Intelligent Autonomous Systems, ICoIAS 2020*, pp. 51–55, 2020, doi: 10.1109/ICoIAS49312.2020.9081833.

[22] S. Maji and S. Bose, "CBIR Using Features Derived by Deep Learning," *ACM/IMS Transactions on Data Science*, vol. 2, no. 3, pp. 1–24, 2021, doi: 10.1145/3470568.

[23] F. A. Putra *et al.*, "Classification of Batik Authenticity Using Convolutional Neural Network Algorithm with Transfer Learning Method," 2021. doi: 10.1109/ICIC54025.2021.9632937.

[24] A. H. Rangkuti, V. H. Athala, N. F. Luthfi, S. V. Aditama, M. M. Ramadhan, and A. H. Aslamia, "Enhancement of traditional clothes pattern recognation using Convolutional Neural Network," in *2021 IEEE International Conference on Computing, ICOCO 2021*, 2021, pp. 224–229. doi: 10.1109/ICOCO53166.2021.9673549.

[25] Z. Widyantoko, T. P. Widowati, Isnaini, and P. Trapsiladi, "Expert role in image classification using cnn for hard to identify object: Distinguishing batik and its imitation," *IAES International Journal of Artificial Intelligence*, vol. 10, no. 1, pp. 93–100, 2021, doi: 10.11591/ijai.v10.i1.pp93-100.

[26] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, 2018, doi: 10.1109/CVPR.2018.00474.

[27] D. M. S. Arsa and A. A. N. H. Susila, "VGG16 in Batik Classification based on Random Forest," in *Proceedings of 2019 International Conference on Information Management and Technology, ICIMTech 2019*, 2019, pp. 295–299. doi: 10.1109/ICIMTech.2019.8843844.

[28] A. Ahmed, "Pre-trained CNNs Models for Content based Image Retrieval," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 7, pp. 200–206, 2021, doi: 10.14569/IJACSA.2021.0120723.

[29] A. Y. Alfakih, Euclidean Distance Matrices and Their Applications in Rigidity Theory. 2018. doi: 10.1007/978-3-319-97846-8.

[30] F. Bimantoro, A. A. Aziz, A. Y. Husodo, A. Musnansyah, A. E. Minarno, and A. Kurniawardhani, "Image Retrieval using Modified Multi Texton and Rotation Invariant Local Binary Pattern," *2020 International Conference on Advancement in Data Science, E-Learning and Information Systems, ICADEIS 2020*, Oct. 2020, doi: 10.1109/ICADEIS49811.2020.9277462.

[31] A. Latif *et al.*, "Content-based image retrieval and feature extraction: A comprehensive review," *Math Probl Eng*, vol. 2019, 2019, doi: 10.1155/2019/9658350.