Modified Convolutional Neural Network Architecture for Batik Motif Image Classification

Ardian Yusuf Wicaksono[1], Nanik Suciati[1], Chastine Fatichah[1], Keiichi Uchimura[2], Gou Koutaki [2]

***Abstract*—Batik is one of the cultural heritages of Indonesia that have many different motifs in each region as well as in its usage. However, the Indonesians sometimes not knowing the batik motif that they're wearing every day, and sometimes they have a batik image without knowing batik information contained in their batik image. With the growing number of images of batik and batik motifs, a classification method that can classify various motifs of batik is required to automatically detect the motif from the batik image. Image processing using the Deep Learning especially for image classification is widely used recently because it has good results. The most popular method in deep learning is Convolutional Neural Network (CNN) which has been proved robust in natural images. This study offers a batik motif image classification system using CNN method with new network architecture developed by combining GoogLeNet and Residual Networks named IncRes. IncRes merges the Inception Module with Residual Network structure. With the 70.84% accuracy, the system can be used to classify the batik image motif accurately.**

***Keywords*— Batik, Classification, Convolutional Neural Network, Deep Learning, Machine Learning, Pattern Recognition.**

I. INTRODUCTION

Batik is one of the traditional fabric that have style, color and texture that represent the cultural intellectual wealth of Indonesia. The batik fabric also recognized as cultural heritage of Indonesia by UNESCO (Masterpieces of the Oral and Intangible Heritage of Humanity) since 2009. The style, motifs and colors of batik are also influenced by the culture from outside Indonesia, such as the influence of Hindu culture, Islam, The Netherlands, China and Japan. There are over 181 batik in Indonesia spread in every region [1], this amount does not include the development of a variety of local batik motifs constantly evolving.

The rapid development of batik motif also affect in the increases number of batik image. With internet is easily accessed by the citizen and smartphone is widely used in Indonesia, the batik image is very easy to get taken by citizen's smartphone camera. The development of batik online store also make batik image number growing quickly. However, many Indonesian people still do not know the types of batik in Indonesia and how to distinguish the motif of the other batik, because in each motif has a meaning and use of culturally diverse in everyday life.

Research in the batik motif recognition has been done by several researchers to get a new method that has better performance than previous studies.

Research of batik image classification has previously been done by [2] using Curvelet Transformation, HSV color space, and k-Nearest Neighbor (KNN) classifier. Researcher [3] also did a comparison method for batik classification using GLCM, Canny edge detection, and Gabor filter. Research by [4] was using Wavelet Transformation and decision tree to optimize the parameters in batik image retrieval. Use of Wavelet Transformation was also done by [5] with the Neural Network as the classifier. Batik Researchers [6] using the Color Co-occurrence Matrix (CCM), Different Between Pixels of Scan Pattern (DBPSP), and Color Histogram for K-Means (CHKM) as a color-texture feature and Backpropagation Neural Network as the classifier. Another study using BPNN has been done by [7] using the Gray Level Co-occurrence Matrix (GLCM) feature and RGB color statistical features. Research from [8] using Local Binary Pattern that invariant with rotation and Probabilistic Neural Network (PNN) as a classifier. Use of PNN is also done with the GLCM feature on research conducted by [9] and [10] which in the second study they used the Multi Texton Histogram features, PNN also used in [11] combining the texture, shape and color feature. With the popularity of Scale invariant Feature Transform (SIFT) methods, SIFT was also used in the batik classification by [12]. SIFT is also used in the study by [13] combined with the Bag of Features (BOF) and Support Vector Machine (SVM) classifier. Research from [14] also use SIFT Features Moments method classified using KNN. From the whole previous research on batik image classification, most of them still use the handcrafted features or feature set manually by researchers, thereby reducing the robustness of the method to the new data, and also the dataset used by previous studies was too small that can't represent the number of batik images that exist today.

The development of deep learning method especially Convolutional Neural Network (CNN) is currently being

[1]Ardian Yusuf Wicaksono, Nanik Suciati and Chastine Fatichah are with Departement of Informatics, Institut Teknologi Sepuluh Nopember, Surabaya, 60111, Indonesia. E-mail: ardianyusufw@gmail.com; nanik@if.its.ac.id; chastine@cs.its.ac.id

[2]Keiichi Uchimura and Gou Koutaki are with Department of Computer Science and Electrical Engineering, Kumamoto University, Kumamoto, 860-8555, Japan. E-mail: uchimura@cs.kumamoto-u.ac.jp; koutaki@cs.kumamoto-u.ac.jp.

intensively used by researchers because it can beat the performance of the method still uses handcrafted features such as SIFT. Developed in early 90's by [15] and [16] for digit and document recognition respectively. The CNN also applied in natural images. As in ImageNet Large Scale Visual Recognition Challenge (ILSVRC), a big scale image classification challenge, the CNN method is dominating as the winner in recent years starting in 2012 by [17]. In 2014, the winner was GoogleNet developed by [18] and the winner of 2015 was Deep Residual Networks by [19]. Usage of CNN in another data also successfully applied in chart dataset and also fish dataset by [20] and [21] respectively.

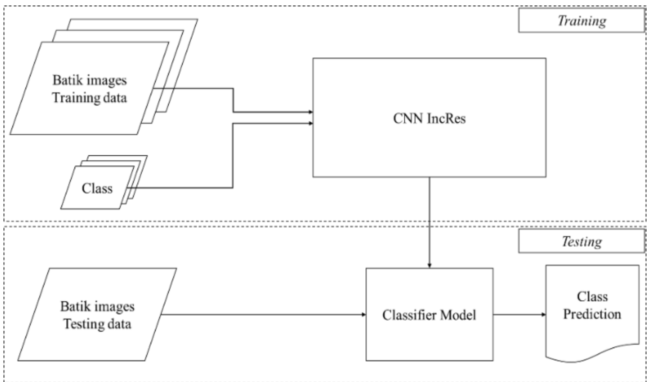

Figure 1. Batik classification system with CNN.

Using the vastly growing number of the batik image and very good performance of recent CNN architecture in most image data, this research proposed a CNN architecture that combine the architecture from GoogLeNet and Residual Network for batik image classification.

II. Method

The system is designed to solve the batik image classification problem using Convolutional Neural Network (CNN). Batik image classification system offered in this study had a design as shown in Figure 1 with two major phases, the first is training phase, this phase is training the system to get a model that is used to classify images and the second one is testing phase, this phase is used to test the trained model using new data outside the training data. In the training phase, the data and their class label used as input in the proposed CNN system with the combination of GoogLeNet and Residual Network architecture that in this study will be named CNN IncRes. From the training results will be obtained classification models to be used in the testing phase. Data testing used as input into the model and obtained class predictions.

The proposed method is using combination of GoogLeNet and Residual Network at module level. The base of the architecture is using the GoogLeNet architecture that have Inception module. In the proposed architecture, inside of Inception module from GoogLeNet, the Residual module is implemented in each of convolutional branch. The overall shape of the proposed network refers to the GoogLeNet with Inception module replaced with proposed IncRes module and also some changes in the amount of output for each layer. The outline of the proposed network is shown here. The network will be preceded by several convolutional layers and max pooling layer to process the input image and then enter to the IncRes module. Multiple IncRes module will be stacked and at the end is using fully connected layer and softmax layer to do the classification. The proposed network architecture can be seen in Figure .

In each IncRes module, each branch doing dimensional reduction in input channel to adjust the number of output on each branch by using 1×1 convolution. After the dimensional reduction in each branch, then enter the convolution process with filter size on each branch is 1×1, 3×3 and 5×5 respectively. In the branch with max pooling, after the max pooling process is do the convolutional 1x1 again as projection of the output. After all convolution and max pooling in each branch, the result in each branch will be added with the shortcut from the output in each 1×1 convolutional dimension reduction. After all the shortcut adding, the output in each branch will be merged or concat. And the concat output is the output of IncRes module. The IncRes module can be seen in.

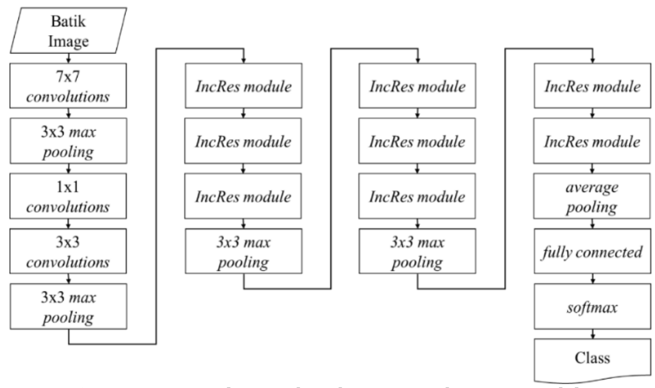

Figure 2. Proposed network architecture with IncRes module.

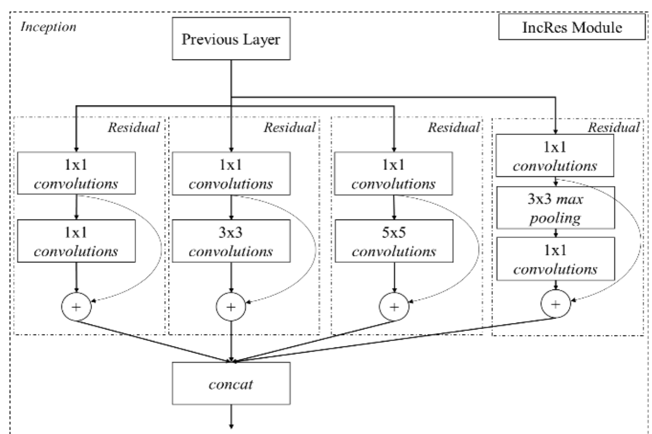

Figure 3. IncRes module detail.

III. Result and Analysis

The dataset used in this research is the batik image photos directly captured from batik cloth, scanned from books and the internet. Total of the data are 7112 batik image with the size 256×256 pixels divided in 11 class batik motif. 6401 images are used in the training process, and the remaining 711 images are for testing process. The image count detail in each class can be seen in Table 1. The class is based from the batik motif in [22] and some new type batik motif.

TABLE 1.
DATASET DETAIL

Class name	Total	Test	Train
buketan	2493	249	2244
ceplok	672	67	605
kawung	128	13	115
kewan	1266	127	1139
lereng	595	60	535
megamendung	434	43	391
parang	221	22	199
pinggiran	342	34	308
sekarjagad	330	33	297
semen	544	54	490
sidomukti	87	9	78
Total	7112	711	6401

This research will compare the result of the GoogLeNet, Residual Network and some variant of the proposed IncRes architecture. Deep learning framework used to implement our model in this experiment is Caffe [23]. The comparison parameter is with accuracy and execution time. The training conducted in here is using random weight initialization and with the same epoch and learning rate. All experiments were conducted with 0,001 learning rate, momentum 0.9, gamma 0.96, stepsize 10000, and the maximum iteration adjusted for the number of batches at each epoch architecture with the same number (500 epochs).

The GoogLeNet v3 and Residual Network with 50 layers are used in this experiment, compared with the variant of the IncRes. In this experiment, there are 4 variant of the IncRes, each variant using the two initial convolutions before the module like in the GoogLeNet, the difference of each variant is just the number of the output channel in each convolution in the each module. The first variant (IncRes-1) is using the number from GoogLeNet architecture. The second variant (IncRes-2) is using the total number each module from Residual Network with 34-layer architecture. The third variant (IncRes-3) is similar with second variant (IncRes-2) but with the two first initial convolution before the module using a different value, increased to 128 and 256 from 64 and 192 like seen in the GoogLeNet. The fourth variant (IncRes-4) is similar with second variant (IncRes-2) but with decreased count of the module so the network is shorter.

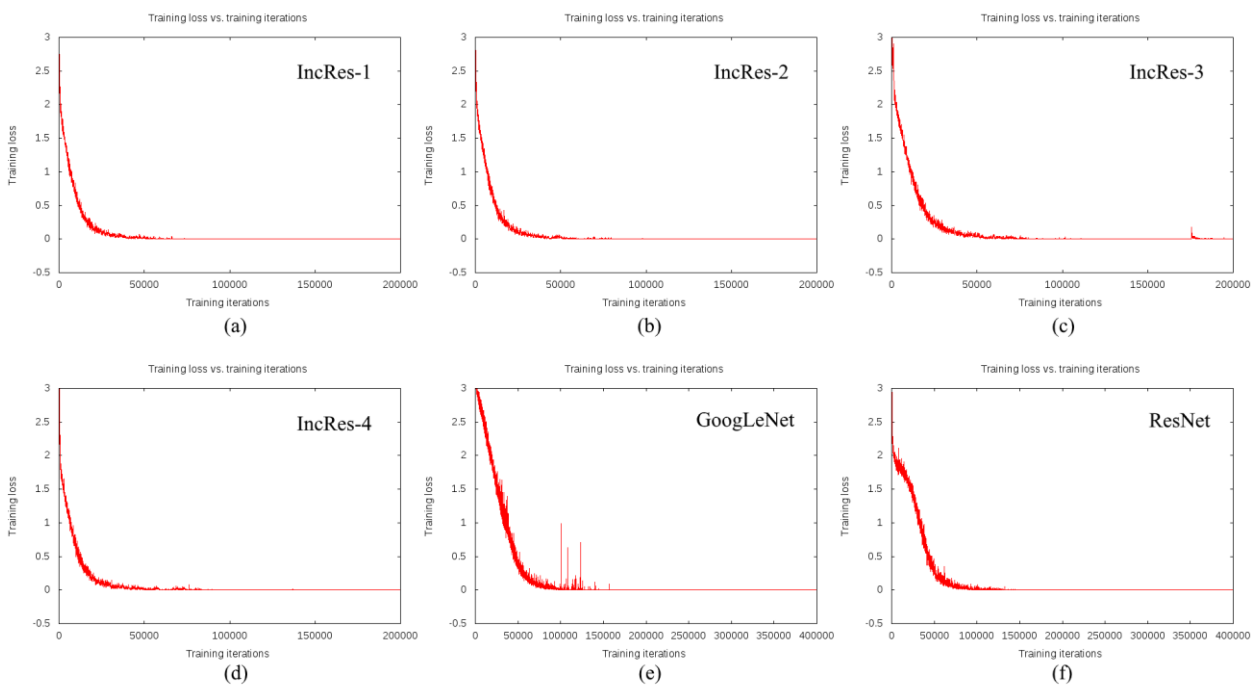

Figure 4. Training loss log

The training loss log can be seen in **Figure** , with IncRes-1 in **Figure** a, IncRes-2 in **Figure** b, IncRes-3 in **Figure** c, IncRes-4 in **Figure** d, GoogLeNet in **Figure** e, and Residual Network in **Figure** f. From the training experiment loss log figure shows the training loss were in a similar declining trend and begin converging in iteration 50000 and fully convergent at iteration 100000, on all four IncRes architectures are only architectural IncRes-3 a little longer convergent, while the other IncRes architecture shows exactly the same trend. In the GoogLeNet and Residual Network, it takes a smaller batch size due to adjusting the size of VRAM in GPU because of larger computing. Therefore, both architecture begin to converge on iteration 100000 and fully convergent at iteration 200000. But, all experiment use the same epoch so the trends are similar to each other.

The testing result of accuracy and execution time per one image can be viewed in **Table**. **Table** shows the IncRes architecture has better accuracy than GoogLeNet and Residual Network with the best accuracy 70.84% by IncRes-2, so the IncRes-2 architecture is selected as the proposed IncRes architecture, this architecture module filter size and output detail can be seen in **Table** . This architecture exceeds the accuracy of Residual Network with a significant distance and bit higher than GoogLeNet. From the time comparison, IncRes-4 architecture have the fastest execution time of 636 ms for one image, it has fastest time execution because it requires fewer computing from the reduction in the number of IncRes module, whereas IncRes-2 architecture that has the highest accuracy is not so much different computing time at 733 ms per image. Can be seen from the **Table** too that the increased size of the output at the initial convolution in IncRes-3 affect the longer computation time, and also with a greater amount of number of output channels on Inception GoogLeNet which is adapted to the IncRes-1 architecture also provides a longer time, but the both architecture has lower accuracy than IncRes-2 architecture. The GoogLeNet and Residual Network has slower execution time because both requires more computation time with more than 0.8 seconds for every single image on GoogLeNet and more than 2.2 seconds to Residual Network 50 layer that indeed does have a larger network size.

IV. CONCLUSION AND FUTURE WORK

The performance of the CNN batik image classification system is very dependent on how well the network is designed to recognize the pattern for the batik data. The proposed CNN method with IncRes network architecture can be used for batik motif image classification with the accuracy 70.84% and computational time 733 ms better than previous method.

TABLE 2.
TESTING RESULT

Architecture	Accuracy (%)	Time (ms)
IncRes-1	68.31	1056
IncRes-2	**70.84**	733
IncRes-3	69.63	990
IncRes-4	68.72	**636**
GoogLeNet	66.94	877
Residual Network	54.25	2228

TABLE 3.
INCRES-2 MODULE FILTER SIZE AND OUTPUT DETAIL

type	filter size/ stride	output size	depth	#1×1 reduce	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool reduce	pool proj
convolution (1)	7×7/2	128×128×64	1								
max pool	3×3/2	64×64×64	0								
convolution (2)	3×3/1	64×64×192	2			64	192				
max pool	3×3/2	32×32×192	0								
incres (3a)		32×32×256	2	64	64	128	128	32	32	32	32
incres (3b)		32×32×256	2	64	64	128	128	32	32	32	32
incres (3b)		32×32×256	2	64	64	128	128	32	32	32	32
max pool	3×3/2	16×16×256	0								
incres (4a)		16×16×512	2	128	128	256	256	64	64	64	64
incres (4b)		16×16×512	2	128	128	256	256	64	64	64	64
incres (4c)		16×16×512	2	128	128	256	256	64	64	64	64
max pool	3×3/2	8×8×512	0								
incres (5a)		8×8×1024	2	256	256	512	512	128	128	128	128
incres (5b)		8×8×1024	2	256	256	512	512	128	128	128	128
avg pool	7×7/1	1×1×1024	0								
connected		1×1×11	1								
softmax		1×1×11	0								

In the future, the architecture will be designed more efficiently to save the computational time, and also usage of transfer learning or fine-tuning in convolution weight from the other better dataset can be applied to get more performance and robustness.

ACKNOWLEDGEMENT

This acknowledgement is aimed to Kementerian Ristek Dikti for giving the BPPDN Fresh Graduate scholarship for taking the master degree at Institut Teknologi Sepuluh Nopember and JASSO for exchange program at Kumamoto University to the first author.

REFERENCE

[1] J. Achjadi, *Batik : spirit of Indonesia*. Yayasan Batik Indonesia, 1999.

[2] N. Suciati, A. Kridanto, M. F. Naufal, M. Machmud, and A. Y. Wicaksono, "Fast discrete curvelet transform and HSV color features for batik image clansificotlon," in *2015 International Conference on Information & Communication Technology and Systems (ICTS)*, 2015, pp. 99–104.

[3] I. Nurhalida, R. Manurung, and A. M. Arymurthy, "Performance comparison analysis features extraction methods for Batik recognition," in *2012 International Conference Advanced Computer Science and Information Systems*, 2012.

[4] A. H. Rangkuti, Z. E. Rasjid, and D. J. Santoso, "Batik Image Classification Using Treeval and Treefit as Decision Tree Function in Optimizing Content Based Batik Image Retrieval," *Procedia Comput. Sci.*, vol. 59, pp. 577–583, 2015.

[5] B. Arisandi, N. Suciati, and A. Yudhi Wijaya, "Pengenalan Motif Batik Menggunakan Rotated Wavelet Filte dan Neural Network," *JUTI, J. Ilm. Teknol. Inf.*, vol. 9, no. 2, pp. 13–19, 2011.

[6] N. Suciati, W. A. Pratomo, and D. Purwitasari, "Batik Motif Classification Using Color-Texture-Based Feature Extraction and

Backpropagation Neural Network," in *2014 IIAI 3rd International Conference on Advanced Applied Informatics*, 2014, pp. 517–521.

[7] C. S. K. Aditya, M. Hani'ah, R. R. Bintana, and N. Suciati, "Batik classification using neural network with gray level co-occurence matrix and statistical color feature extraction," in *2015 International Conference on Information & Communication Technology and Systems (ICTS)*, 2015, pp. 163–168.

[8] A. Kurniawardhani, N. Suciati, and I. Arieshanti, "Klasifikasi Citra Batik Menggunakan Metode Ekstraksi Ciri yang Invariant Terhadap Rotasi," *JUTI J. Ilm. Teknol. Inf.*, vol. 12, no. 2, p. 48, Jul. 2014.

[9] A. E. Minarno, Y. Munarko, A. Kurniawardhani, F. Bimantoro, and N. Suciati, "Texture feature extraction using co-occurrence matrices of sub-band image for batik image classification," in *2014 2nd International Conference on Information and Communication Technology (ICoICT)*, 2014, pp. 249–254.

[10] A. E. Minarno, Y. Munarko, A. Kurniawardhani, and F. Bimantoro, "Classification of Texture Using Multi Texton Histogram and Probabilistic Neural Network," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 105, no. 1, p. 12022, Jan. 2016.

[11] A. Nilogiri, "Klasifikasi Kansei Multi Label dengan Probabilistic Neural Network pada Citra Batik menggunakan Kombinasi Fitur Warna, Tekstur, dan Bentuk," Institut Teknologi Sepuluh Nopember, 2012.

[12] I. Nurhaida, A. Noviyanto, R. Manurung, and A. M. Arymurthy, "Automatic Indonesian's Batik Pattern Recognition Using SIFT Approach," *Procedia Comput. Sci.*, vol. 59, pp. 567–576, 2015.

[13] R. Azhar, D. Tuwohingide, D. Kamudi, Sarimuddin, and N. Suciati, "Batik Image Classification Using SIFT Feature Extraction, Bag of Features and Support Vector Machine," *Procedia Comput. Sci.*, vol. 72, pp. 24–30, 2015.

[14] I. Setyawan, I. K. Timotius, and M. Kalvin, "Automatic batik motifs classification using various combinations of SIFT features moments and k-Nearest Neighbor," in *2015 7th International Conference on Information Technology and Electrical Engineering (ICITEE)*, 2015, pp. 269–274.

[15] Y. LeCun *et al.*, "Handwritten Digit Recognition with a Back-Propagation Network," in *Advances in Neural Information Processing Systems 2*, D. S. Touretzky, Ed. Morgan-Kaufmann, 1990, pp. 396–404.

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

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.

[18] C. Szegedy *et al.*, "Going Deeper with Convolutions," *CoRR*, vol. abs/1409.4, 2014.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *CoRR*, vol. abs/1512.0, 2015.

[20] B. Tang *et al.*, "DeepChart: Combining deep convolutional networks and deep belief networks in chart classification," *Signal Processing*, vol. 124, pp. 156–161, Jul. 2016.

[21] H. Qin, X. Li, J. Liang, Y. Peng, and C. Zhang, "DeepFish: Accurate underwater live fish recognition with a deep architecture," *Neurocomputing*, vol. 187, pp. 49–58, 2016.

[22] S. K. S. Susanto, *Seni kerajinan batik Indonesia*. Jakarta: Balai Penelitian Batik dan Kerajinan, Lembaga Penelitian dan Pendidikan Industri, Departemen Perindustrian R.I., 1973.

[23] Y. Jia *et al.*, "Caffe: Convolutional Architecture for Fast Feature Embedding," *CoRR*, vol. abs/1408.5, 2014.