

## Implementasi Data Augmentation Random Erasing dan GridMask pada CNN untuk Klasifikasi Batik

### *Implementation of Random Erasing and GridMask Data Augmentations on CNN for Batik Classification*

Chan Uswatun Khasanah<sup>\*1</sup>, Angkin Kusuma Pertiwi<sup>2</sup>, Farrel Witamajaya<sup>3</sup>  
<sup>1,2,3</sup>Politeknik Akbara Surakarta; Jl. Sumbing Raya, Mojosongo, Jebres, Surakarta,  
(0271) 8591185  
e-mail: \*[1chanuswatunkhasanah@gmail.com](mailto:chanuswatunkhasanah@gmail.com), [2angkinkusuma123@gmail.com](mailto:angkinkusuma123@gmail.com),  
[3farrel.witamajaya@gmail.com](mailto:farrel.witamajaya@gmail.com)

#### **Abstrak**

Penelitian ini bertujuan untuk mengetahui pengaruh data augmentation Random Erasing dan GridMask pada klasifikasi batik dengan 550 gambar yang terbagi menjadi 5 kelas, yaitu Ceplok, Kawung, Lereng, Nitik, dan Parang, serta mengetahui manakah data augmentation yang memiliki akurasi lebih tinggi. Dataset terbagi menjadi data train, validation, dan test dengan perbandingan 70% : 20% : 10% sehingga masing-masing data train dan validation terdiri dari 500 gambar dan jumlah data test adalah 50 gambar. Kami mengimplementasikan data augmentation Random Erasing dan GridMask pada model pre-trained VGG16 dengan metode transfer learning dan fine-tuning (melakukan training pada block5 convolutional layer). Berdasarkan training menggunakan model pre-trained VGG16 pada dataset batik dengan membandingkan metode fine-tuning dan transfer learning menunjukkan bahwa metode fine-tuning menghasilkan akurasi lebih tinggi daripada transfer learning. Training tanpa data augmentation dapat menghasilkan akurasi yang lebih tinggi daripada saat menggunakan data augmentation, namun jumlah gambar yang diprediksi benar lebih sedikit (mengalami overfitting). Meningkatkan akurasi dan memperkecil overfitting dapat dilakukan dengan mengimplementasikan data augmentation Random Erasing dan GridMask. Dari kedua data augmentation tersebut yang memiliki akurasi paling tinggi pada klasifikasi batik menggunakan CNN adalah GridMask. Kendala penelitian ini adalah kurva pelatihan yang dihasilkan masih belum cukup stabil (fluktuatif) karena menggunakan dataset yang kecil dan gambar-gambar batik pada dataset tersebut belum bisa merepresentasikan berbagai macam motif batik. Di masa mendatang, peneliti berharap adanya dataset batik yang sudah tersedia secara "official" dengan jumlah gambar yang banyak dan kompleks sehingga penelitian-penelitian klasifikasi batik dapat menggunakan dataset tersebut.

**Kata kunci**—Klasifikasi batik, VGG16, transfer learning, fine-tuning, overfitting, data augmentation

#### **Abstract**

This study aims to determine the effect of Random Erasing and GridMask data augmentations on the classification of batik with 550 images divided into 5 classes, namely Ceplok, Kawung, Slope, Nitik, and Parang, and to find out which data augmentation has higher accuracy.. The dataset is divided into train, validation, and test data with a ratio of 70% : 20% :

10% so that each train and validation data consists of 500 images and the number of test data is 50 images. We implemented Random Erasing and GridMask data augmentations on the pre-trained VGG16 model with transfer learning and fine-tuning methods (training on the block5 convolutional layer). Based on training using the VGG16 pre-trained model on the batik dataset by comparing the fine-tuning and transfer learning methods, it shows that the fine-tuning method produces higher accuracy than transfer learning. Training without data augmentation can produce higher accuracy than when using data augmentation, but the number of images that are correctly predicted is less (overfitting). Improving accuracy and minimizing overfitting can be done by implementing Random Erasing and GridMask data augmentations. Data augmentations with highest accuracy in the classification of batik using CNN is GridMask. The problem of this research is that the resulting training curve is still unstable (fluctuative) because it uses a small dataset and the batik images in this dataset cannot represent various kinds of batik motifs. In the future, researchers hope that there will be an "official" available batik dataset with a large number of complex images so that batik classification studies can use this dataset.

**Keywords**—Batik classification, VGG16, transfer learning, fine-tuning, overfitting, data augmentation

## 1. PENDAHULUAN

*Image processing* adalah bidang penelitian yang sangat aktual dan menantang serta menjadi bagian dari kehidupan sehari-hari. Convolutional Neural Network (CNN) dikenal berhasil memecahkan masalah yang berhubungan dengan citra/gambar. CNN memiliki kemampuan untuk mendeteksi dan mengenali objek pada sebuah gambar serta dapat mengungguli metode tradisional dalam tugas *computer vision* dan *pattern recognition*, seperti *object detection* [1], *classification* [2], *image segmentation* [3], [4], dan *text recognition* [5].

CNN sudah banyak diimplementasikan pada berbagai macam pekerjaan, salah satunya adalah klasifikasi batik. Tristanto dkk [6] dan Handhayani dkk [7] melakukan klasifikasi batik dengan merancang CNN menggunakan empat *convolutional layer* dan tiga *convolutional layer*. Selain menggunakan arsitektur CNN yang dirancang sendiri, beberapa penelitian juga melakukan klasifikasi batik menggunakan model *pre-trained*, seperti VGG16 dan VGG19 [8], [9] serta IncRes [10]. VGG16 berhasil mengungguli metode SURF dan SIFT dalam klasifikasi batik [9] dengan rata-rata akurasi  $89\% \pm 7\%$ , sedangkan model berbasis SIFT dan SURF mencapai rata-rata akurasi  $88 \pm 10\%$  dan  $88 \pm 8\%$ . Karena alasan tersebut, CNN menjadi salah satu metode yang dapat digunakan untuk klasifikasi dengan hasil yang optimal.

Kemampuan untuk menggeneralisasi adalah hal yang harus diperhatikan pada CNN. Ketika suatu model sangat kompleks, seperti memiliki banyak parameter dibandingkan jumlah sampel pelatihan, maka *overfitting* dapat terjadi dan melemahkan kemampuan generalisasi. *Overfitting* adalah suatu keadaan saat model belajar dengan baik terhadap data *train*, namun tidak dapat memprediksi dengan tepat saat diberikan data lain yang serupa [11].

Krizhevsky dkk [12] melakukan *data augmentation* dan *dropout* untuk mengatasi *overfitting*. *Data augmentation* dipilih karena memiliki kemampuan meningkatkan variasi gambar dengan melakukan manipulasi melalui beberapa cara, seperti *flipping*, *resizing*, *random-cropping*, dan *color-translating* [13] sehingga cocok apabila diimplementasikan pada *dataset* kecil. Beberapa penelitian juga telah memperkenalkan teknik *data augmentation* baru, seperti Random Erasing [14] dan GridMask [15] yang telah diuji coba pada *dataset*, seperti CIFAR10 dan CIFAR100 serta pada arsitektur CNN yang memiliki beragam parameter [16]–[18].

Berdasarkan uraian di atas, penelitian akan mengimplementasikan *data augmentation* Random Erasing dan GridMask pada model *pre-trained* VGG16 dengan metode *fine-tuning* dan *transfer learning* untuk melakukan klasifikasi batik. Pada metode *fine-tuning*, pelatihan hanya

---

# Implementasi Data Augmentation Random Erasing dan GridMask pada CNN untuk Klasifikasi Batik

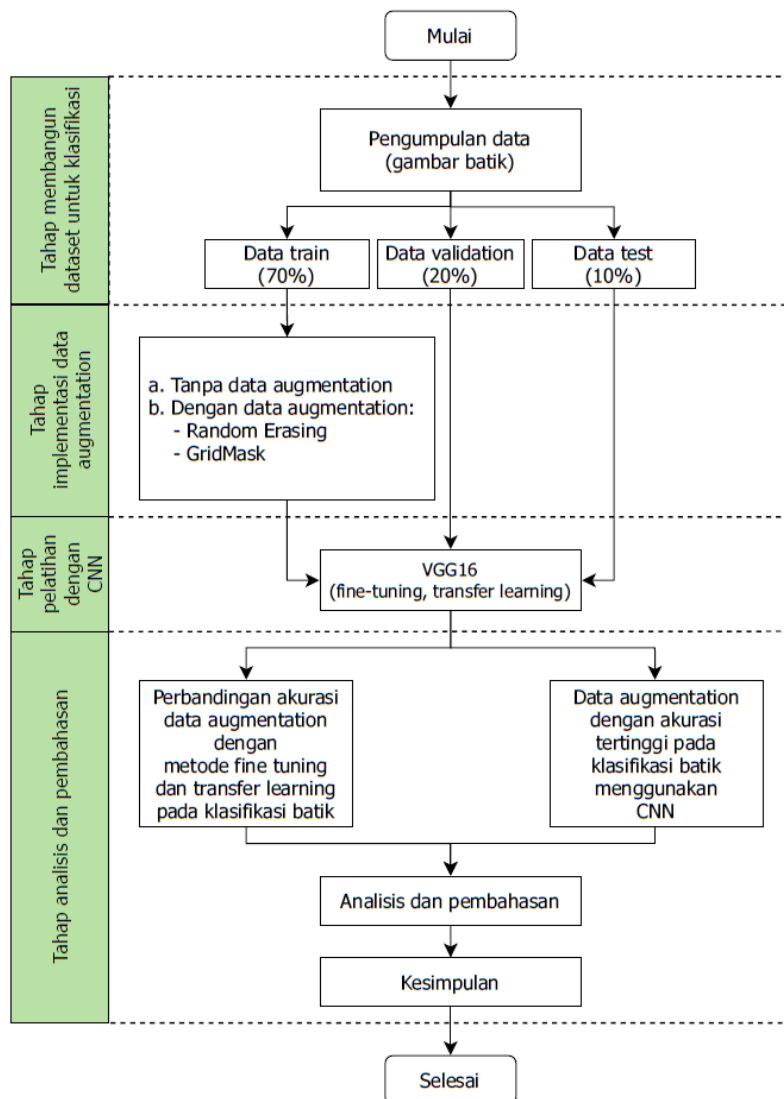
dilakukan pada *convolutional layer block5 (trainable layer)*. Dataset yang digunakan adalah 550 gambar batik yang terdiri dari 5 kelas, yaitu Ceplok, Kawung, Lereng, Nitik, dan Parang. Jumlah data tersebut tergolong kecil untuk digunakan pada CNN sehingga memiliki peluang lebih besar terjadi *overfitting*. Peneliti menerapkan *data augmentation* untuk melihat efek apa yang dihasilkan pada klasifikasi batik berdasarkan nilai akurasi.

## 2. METODE PENELITIAN

### 2.1 Metode Pengumpulan Data

Pengumpulan data dilakukan untuk mendapatkan berbagai macam data yang diperlukan selama penelitian. Data yang digunakan diunduh di Google Image dengan jumlah 300 gambar untuk lima jenis batik [19].

### 2.2 Alur Penelitian



Gambar 1. Alur penelitian

Rincian alur penelitian berdasarkan Gambar 1 adalah sebagai berikut.

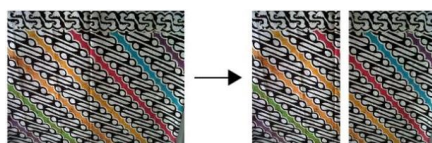
- a. Tahap membangun *dataset*  
*Dataset* berisi 550 gambar batik yang terdiri dari 5 kelas, yaitu Ceplok, Kawung, Lereng, Nitik, dan Parang. *Dataset* diperoleh dengan mengunduh gambar batik di Google Image dengan menuliskan *keyword*, seperti “batik Ceplok” (tanpa tanda petik). Tidak ada kriteria khusus dalam menentukan gambar batik yang digunakan untuk *dataset*, seperti ukuran, warna, jenis, dan sebagainya. *Dataset* terbagi menjadi tiga bagian, yaitu 70% data *train*, 20% data *validation*, dan 10% data *test*. Data *train* dan *validation* digunakan untuk melatih model CNN pada klasifikasi batik, sedangkan data *test* digunakan untuk memprediksi gambar batik.
- b. Tahap implementasi *data augmentation*  
Terdapat dua skenario pada penggunaan *data augmentation*, yaitu data *train* tanpa mengimplementasikan *data augmentation* dan data *train* dengan mengimplementasikan *data augmentation*. *Data augmentation* yang digunakan adalah Random Erasing dan GridMask.
- c. Tahap pelatihan menggunakan CNN  
Model CNN yang digunakan untuk pelatihan adalah VGG16 [8], [9].
- d. Tahap analisis dan pembahasan  
Tahap ini merupakan tahap untuk melakukan evaluasi (penilaian) berdasarkan tahap pelatihan menggunakan CNN. Ada dua evaluasi yang dilakukan, yaitu 1) Perbandingan akurasi *data augmentation* dengan metode *fine-tuning dan transfer learning* [8] pada klasifikasi batik, 2) *Data augmentation* dengan akurasi tertinggi pada klasifikasi batik

### 3. HASIL DAN PEMBAHASAN

#### 3.1 Tahap Membangun Dataset

##### 3.1.1 Pengumpulan Data

Data yang digunakan berupa gambar yang terdiri dari lima macam motif batik, yaitu Ceplok, Kawung, Lereng, Nitik, dan Parang. Gambar diperoleh dengan mengunduh dari Google Image dengan rincian 250 gambar untuk data *train* dan *validation* serta 50 gambar untuk data *test*. Kami mengembangkan 250 gambar menjadi 500 gambar menggunakan metode *slicing* dengan membagi gambar menjadi dua bagian yang sama [8], [9], seperti yang ditunjukkan pada Gambar 2.



Gambar 2. Metode *slicing* pada gambar batik

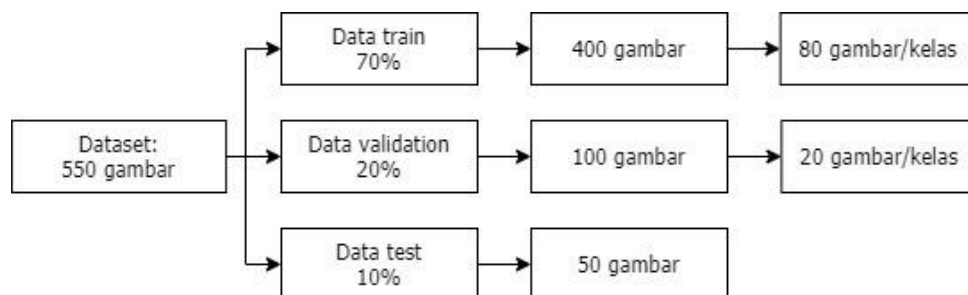
##### 3.1.2 Pengelompokan Data

Setelah data terkumpul, masing-masing kelompok gambar dibagi menjadi data *train*, *validation*, dan *test* dengan struktur *folder* seperti pada Gambar 3.

```
./data/  
  train/  
    ceplok/  
      ceplok1.jpg  
      ceplok2.jpg  
      .....  
    kawung/  
      kawung1.jpg  
      kawung2.jpg  
      .....  
    ...../  
  validation/  
    ceplok/  
      ceplok_a.jpg  
      ceplok_b.jpg  
      .....  
    kawung/  
      kawung_a.jpg  
      kawung_b.jpg  
      .....  
    ...../  
  test/  
    gambar1.jpg  
    gambar2.jpg  
    .....
```

Gambar 3. Struktur folder

Perbandingan jumlah gambar pada data *train*, *validation*, dan *test* adalah 70% : 20% : 10%, seperti yang ditunjukkan pada Gambar 4. Setiap kelas terbagi menjadi 80 gambar untuk data *train* dan 20 gambar untuk data *validation*, sedangkan data *test* terdiri dari 50 gambar.



Gambar 4. Pengelompokan data

## 3.2 Tahap Implementasi Data Augmentation

### 3.2.1 Random Erasing



Gambar 5. Visualisasi Random Erasing

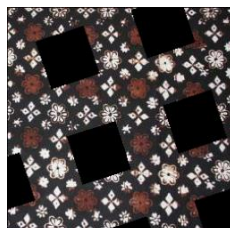
Random Erasing bekerja dengan cara melakukan penghapusan secara acak pada gambar, seperti yang ditunjukkan pada Gambar 5. Cara mengimplementasikan Random Erasing adalah dengan menambahkan *script* di bawah ini pada model pelatihan.

```

1. train_datagen = ImageDataGenerator(
2.     preprocessing_function=get_random_eraser(p=0.5,
3.                                             s_l=0.02, s_h=0.4,
4.                                             r_l=0.3, r_2=1/0.3,
5.                                             v_l=0, v_h=350,
6.                                             pixel_level=False)
7. )

```

### 3.2.2 GridMask



Gambar 6. Visualisasi GridMask

GridMask bekerja dengan cara menambahkan pola *grid* pada gambar. Cara mengimplementasikan GridMask adalah dengan menambahkan *script* di bawah ini pada model pelatihan.

```

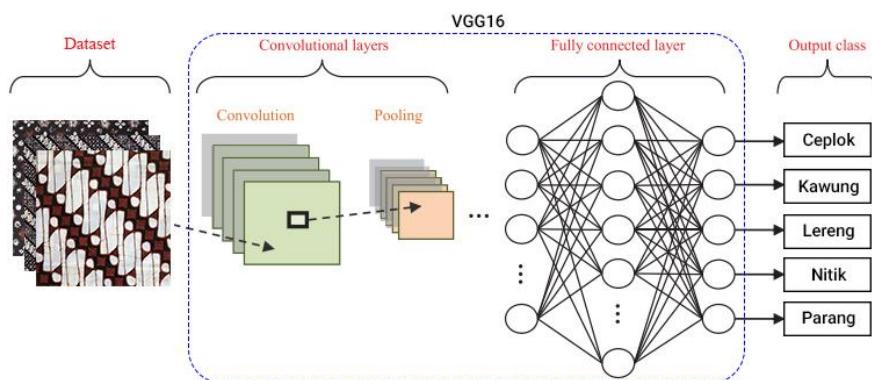
1. transforms_train = albumentations.Compose([
2.     albumentations.OneOf([
3.         GridMask(num_grid=3, mode=0, rotate=180),
4.         GridMask(num_grid=3, mode=2, rotate=180),
5.     ], p=0.7)
6. ])

```

### 3.3 Tahap Pelatihan dengan CNN

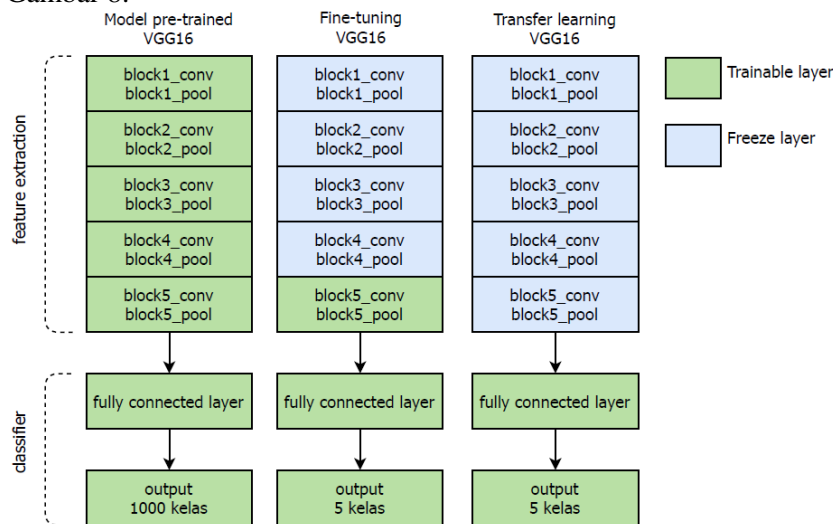
#### 3.3.1 Transfer Learning dan Fine-Tuning VGG16

Kami menggunakan CNN untuk melakukan klasifikasi batik seperti yang ditunjukkan Gambar 7. *Dataset* dilatih menggunakan model CNN, yaitu VGG16 agar model dapat mengklasifikasikan 5 jenis batik, yaitu Ceplok, Kawung, Lereng, Nitik, dan Parang.



Gambar 7. Klasifikasi batik menggunakan CNN

VGG16 terdiri dari 5 *block convolutional layer* sebagai *feature extraction*. VGG16 telah dilatih untuk melakukan klasifikasi pada 1000 kelas dengan *dataset* Imagenet yang terdiri dari 1.200.000 gambar untuk data *train* dan 100.000 gambar untuk data *test*. Kami menggunakan metode *transfer learning* [8], [9] dan *fine-tuning* saat menggunakan VGG16, seperti yang ditunjukkan Gambar 8.



Gambar 8. Transfer learning dan fine-tuning VGG16

*Transfer learning* berarti menggunakan kembali kemampuan klasifikasi VGG16 pada *dataset* yang berbeda (*dataset* batik). Untuk melakukan klasifikasi batik menggunakan *transfer learning* maka *convolutional layer* harus dibekukan (*freeze layer*) dan hanya melakukan pelatihan (*trainable layer*) pada *fully connected layer*. Konsep antara *transfer learning* dan *fine-tuning* hampir sama. Jika menggunakan *fine-tuning* maka beberapa *convolutional layer* ada yang dilatih kembali (*trainable layer*). Pada metode *fine-tuning*, kami melakukan pelatihan hanya pada *convolutional layer block5 (trainable layer)* dan membekukan (*freeze layer convolutional block 1 sampai block 4*). Kami juga merancang *fully connected layer* sesuai dengan model pelatihan yang akan kami buat. Pada bagian *output*, kami membagi menjadi 5 kelas, sesuai dengan jumlah kelas yang kami miliki.

Rincian dari setiap *convolutional layer* pada model *pre-trained VGG16* dengan *input size* 96 px ditunjukkan pada Tabel 1. Baris yang tanpa cetak tebal (*bold*) merupakan *convolutional layer* yang kami bekukan (*freeze layer*), sedangkan baris yang cetak tebal (*bold*) merupakan *convolutional layer* yang kami latih (*trainable layer*).

Tabel 1. Arsitektur Model Pre-Trained VGG16

Layer Name	Type	VGG16	
		Output Shape	Param
input_1	InputLayer	96, 96, 3	0
<b>block1_conv1</b>	<b>Conv2D</b>	<b>96, 96, 64</b>	<b>1792</b>
<b>block1_conv2</b>	<b>Conv2D</b>	<b>96, 96, 64</b>	<b>36928</b>
block1_pool	MaxPooling2D	48, 48, 64	0
<b>block2_conv1</b>	<b>Conv2D</b>	<b>48, 48, 128</b>	<b>73856</b>
<b>block2_conv2</b>	<b>Conv2D</b>	<b>48, 48, 128</b>	<b>147584</b>
block2_pool	MaxPooling2D	24, 24, 128	0
<b>block3_conv1</b>	<b>Conv2D</b>	<b>24, 24, 256</b>	<b>295168</b>

block3_conv2	Conv2D	24, 24, 256	590080
block3_conv3	Conv2D	24, 24, 256	590080
block3_pool	MaxPooling2D	12, 12, 256	0
block4_conv1	Conv2D	12, 12, 512	1180160
block4_conv2	Conv2D	12, 12, 512	2359808
block4_conv3	Conv2D	12, 12, 512	2359808
block4_pool	MaxPooling2D	6, 6, 512	0
<b>block5_conv1</b>	<b>Conv2D</b>	<b>6, 6, 512</b>	<b>2359808</b>
<b>block5_conv2</b>	<b>Conv2D</b>	<b>6, 6, 512</b>	<b>2359808</b>
<b>block5_conv3</b>	<b>Conv2D</b>	<b>6, 6, 512</b>	<b>2359808</b>
<b>block5_pool</b>	<b>MaxPooling2D</b>	<b>3, 3, 512</b>	<b>0</b>

Perbedaan utama antara metode *transfer learning* dan *fine-tuning* yang kami lakukan terletak pada jumlah parameter yang dilatih (*trainable parameter*). Jumlah parameter yang dilatih dengan metode *fine-tuning* lebih banyak daripada metode *transfer learning* dengan selisih hingga tujuh juta parameter, seperti yang ditunjukkan pada Tabel 2.

Tabel 2. *Trainable Parameter*

	<i>Transfer Learning</i> VGG16	<i>Fine-Tuning</i> VGG16
Total parameter	15.895.877	15.895.877
Trainable parameter	1.181.189	8.260.613
Non-trainable parameter	14.714.688	7.635.264

### 3.3.2 Training dengan Model Pre-Trained VGG16

Saat melakukan pelatihan (*training*) menggunakan VGG16, parameter yang kami gunakan ditunjukkan pada Tabel 3.

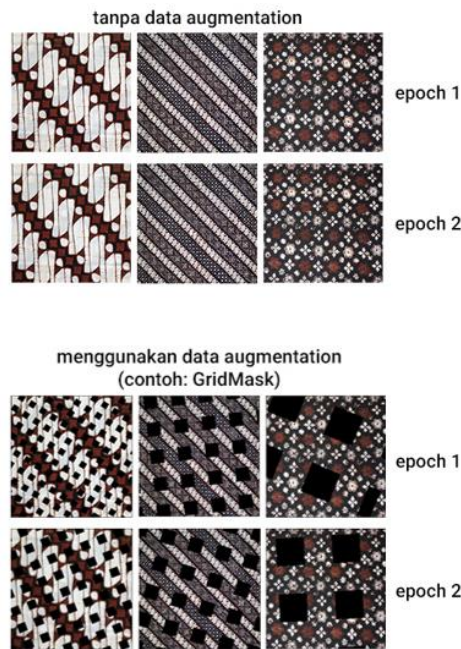
Tabel 3. Parameter Pelatihan

Optimizer	RMSprop
Learning rate	1e-4
Loss	categorical crossentropy
Metrics	accuracy
Epochs	60

Kumpulan gambar pada *dataset* diproses pada setiap *epoch*. Setiap *epoch* memproses gambar yang sama apabila melakukan *training* tanpa menggunakan *data augmentation*. Sebaliknya, setiap *epoch* akan memproses gambar yang “berbeda” jika menggunakan *data augmentation*. Ilustrasi proses *training* menggunakan dan tanpa menggunakan *data augmentation* ditunjukkan pada Gambar 9.

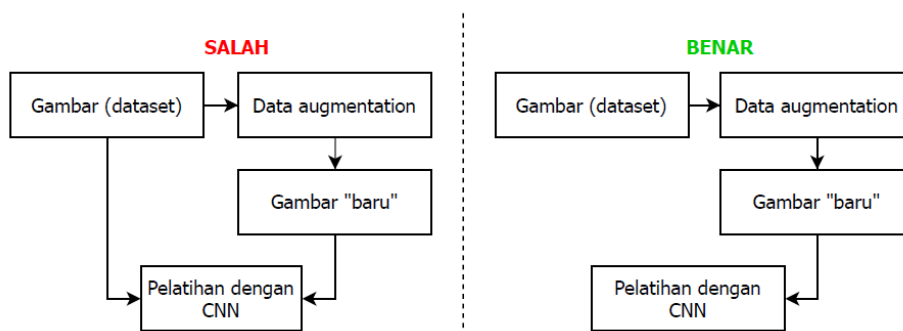


# Implementasi Data Augmentation Random Erasing dan GridMask pada CNN untuk Klasifikasi Batik



Gambar 9. Perbandingan dengan dan tanpa *data augmentation*

Saat menggunakan *data augmentation* maka gambar asli tidak digunakan saat proses *training*, seperti yang ditunjukkan pada Gambar 10. Hal ini bertujuan untuk memastikan bahwa *network* melihat gambar “baru” yang belum pernah “dilihat” sebelumnya di setiap *epoch*. Jika data pelatihan asli (gambar asli sebelum diimplementasikan *data augmentation*) diikutsertakan bersama gambar “baru” maka model akan “melihat” data pelatihan asli beberapa kali. Ingat bahwa tujuan *data augmentation* adalah meningkatkan generalisasi model. Untuk mencapai tujuan tersebut maka data pelatihan “diganti” dengan *data augmentation* yang diubah secara acak.



Gambar 10. Konsep *data augmentation*

## 3.4 Tahap Analisis dan Pembahasan

### 3.4.1 Hasil Implementasi Random Erasing

Random Erasing diimplementasikan dengan empat macam parameter “maksimum area yang terhapus (*v\_h*)”. Nilai “*v\_h*” yang digunakan adalah 50, 150, 250, dan 350, kemudian *training* menggunakan CNN sehingga mendapatkan nilai akurasi seperti yang ditunjukkan pada Tabel 4.

Tabel 4. Nilai Akurasi (%) Random Erasing

Model	Nilai Maksimum Area yang Terhapus (v_h)			
	50	150	250	350
<i>Transfer learning</i> VGG16	81,25	79,17	81,25	<b>82,29</b>
<i>Fine-tuning</i> VGG16	90,63	<b>91,67</b>	88,54	90,63

\*tulisan tebal (*bold*) menunjukkan parameter dengan nilai akurasi tertinggi

### 3.4.2 Hasil Implementasi GridMask

GridMask diimplementasikan dengan empat macam parameter “rotate”. Nilai “rotate” yang digunakan adalah 0, 90, 180, dan 270, kemudian *training* menggunakan CNN sehingga mendapatkan nilai akurasi seperti yang ditunjukkan pada Tabel 5.

Tabel 5. Nilai Akurasi (%) GridMask

Model	Rotate			
	0	90	180	270
<i>Transfer learning</i> VGG16	78,13	<b>79,17</b>	73,96	75
<i>Fine-tuning</i> VGG16	<b>96,88</b>	86,46	93,75	88,54

\*tulisan tebal (*bold*) menunjukkan parameter dengan nilai akurasi tertinggi

### 3.4.2 Hasil

Nilai akurasi pada klasifikasi batik setelah mengimplementasikan *data augmentation* ditunjukkan pada Tabel 6.

Tabel 6. Nilai Akurasi Klasifikasi Batik

Model	Data Augmentation	Parameter	Akurasi (%)	Gambar (data test) yang diprediksi benar
<i>Transfer learning</i> VGG16	Tanpa <i>data augmentation</i>	-	85,42	18
	Random Erasing	v_h = 350	82,29	18
	GridMask	rotate = 90	79,17	23
<i>Fine-tuning</i> VGG16	Tanpa <i>data augmentation</i>	-	91,67	24
	Random Erasing	v_h = 150	91,67	25
	GridMask	rotate = 0	96,88	26

Nilai akurasi pada Tabel 6 merupakan nilai akurasi tertinggi yang didapatkan dari uji coba menggunakan empat parameter yang berbeda di setiap metode *data augmentation*. Berdasarkan Tabel 6, klasifikasi batik menghasilkan akurasi yang lebih tinggi saat menggunakan metode *fine-tuning* daripada *transfer learning* serta saat mengimplementasikan *data augmentation*.

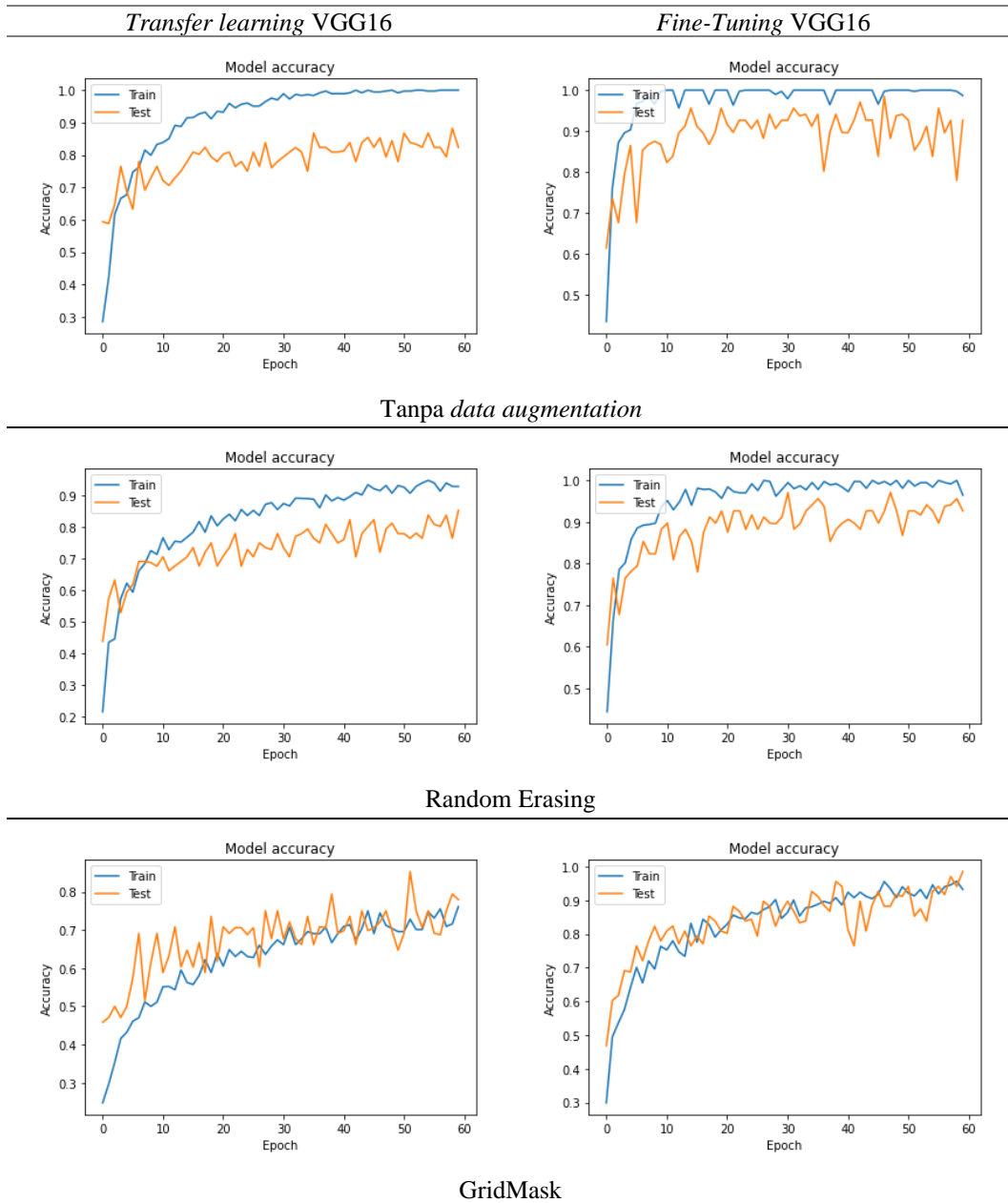
Pada model *transfer learning* VGG16, nilai akurasi tanpa mengimplementasikan *data augmentation* memberikan hasil yang tinggi (85,42%) namun gambar yang diprediksi benar lebih sedikit (18 gambar). Hal inilah yang dinamakan *overfitting*, yaitu suatu keadaan ketika model memiliki kemampuan yang baik saat mempelajari data *train* namun tidak bisa memprediksi dengan tepat jika diberikan data lain (data *test*).

*Data augmentation* dapat digunakan untuk memperkecil *overfitting* meskipun nilai akurasi yang dihasilkan lebih kecil, namun jumlah gambar yang diprediksi benar lebih banyak. Salah satu cara untuk mengetahui apakah model mengalami *overfitting* adalah dengan melakukan pengamatan pada kurva akurasi, seperti yang ditunjukkan pada Tabel 7. Kurva akurasi yang memiliki *gap* cukup jauh antara kemampuan model dalam mempelajari data *train* (garis warna biru) dan memprediksi data *test* (garis warna orange) maka dapat dipastikan model tersebut mengalami *overfitting*.

# Implementasi Data Augmentation Random Erasing dan GridMask pada CNN untuk Klasifikasi Batik

Kurva akurasi pada Tabel 7 masih mengalami fluktuasi (tidak stabil), yaitu keadaan saat tingkat sensitifitas model masih tinggi yang menunjukkan bahwa *overfitting* masih belum dapat diatasi secara maksimal. Tingkat sensitifitas tinggi dikarenakan model belum bisa menggeneralisasi ciri-ciri dari lima macam motif batik akibat dari jumlah *dataset* yang belum kompleks. Hal ini juga dibuktikan dengan rata-rata jumlah gambar yang diprediksi dengan benar hanya berkisar 50% dari 50 gambar pada data *test*. Maka dari itu diperlukan jumlah data yang lebih kompleks dan implementasi *data augmentation* yang lebih “agresif”.

Tabel 7. Kurva Akurasi Klasifikasi Batik



## 4. KESIMPULAN

Berdasarkan tahap analisis dan pembahasan yang telah dilakukan maka dapat disimpulkan bahwa:

- a. *Data augmentation* Random Erasing dan GridMask dapat memperkecil *overfitting* dan meningkatkan akurasi pada klasifikasi batik. Tanpa menggunakan *data augmentation* maka nilai akurasi yang dihasilkan dapat lebih tinggi daripada saat menggunakan *data augmentation*, namun jumlah gambar yang diprediksi benar lebih sedikit (mengalami *overfitting*).
- b. *Data augmentation* Random Erasing dan GridMask yang memiliki akurasi paling tinggi pada klasifikasi batik menggunakan CNN adalah GridMask.

## 5. SARAN

Implementasi *data augmentation* berhasil memperkecil *overfitting* dan meningkatkan akurasi pada klasifikasi batik. Namun kurva pelatihan yang dihasilkan masih belum cukup stabil (fluktuatif) karena menggunakan *dataset* yang kecil dan gambar-gambar batik pada *dataset* tersebut belum bisa merepresentasikan berbagai macam motif batik. Hasil penelitian ini juga belum diimplementasikan secara nyata. Di masa mendatang, peneliti berharap adanya *dataset* batik yang sudah tersedia secara “official” dengan jumlah gambar yang banyak dan kompleks sehingga penelitian-penelitian klasifikasi batik dapat menggunakan *dataset* tersebut. Peneliti juga berharap agar hasil penelitian ini dapat diimplementasikan pada suatu aplikasi/perangkat sehingga *output* penelitian ini dapat terlihat secara nyata.

## DAFTAR PUSTAKA

- [1] R. L. Galvez, A. A. Bandala, E. P. Dadios, R. R. P. Vicerra, and J. M. Z. Maningo, “Object Detection Using Convolutional Neural Networks,” in *IEEE Region 10 Annual International Conference, Proceedings/TENCON*, 2019, vol. 2018-October. doi: 10.1109/TENCON.2018.8650517.
- [2] N. Jmour, S. Zayen, and A. Abdelkrim, “Convolutional neural networks for image classification,” in *2018 International Conference on Advanced Systems and Electric Technologies, IC\_ASET 2018*, 2018. doi: 10.1109/ASET.2018.8379889.
- [3] D. Liao, H. Lu, X. Xu, and Q. Gao, “Image Segmentation Based on Deep Learning Features,” in *11th International Conference on Advanced Computational Intelligence, ICACI 2019*, 2019. doi: 10.1109/ICACI.2019.8778464.
- [4] A. J. Rozaqi, A. Sunyoto, and M. rudyanto Arief, “Deteksi Penyakit Pada Daun Kentang Menggunakan Pengolahan Citra dengan Metode Convolutional Neural Network,” *Creative Information Technology Journal*, vol. 8, no. 1, 2021, doi: 10.24076/citec.2021v8i1.263.
- [5] X. Liu, T. Kawanishi, X. Wu, and K. Kashino, “Scene text recognition with high performance CNN classifier and efficient word inference,” in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2016, vol. 2016-May. doi: 10.1109/ICASSP.2016.7471891.
- [6] J. Tristanto, J. Hendryli, and D. Erny Herwindiati, “Classification of Batik Motifs Using Convolutional Neural Networks,” *SSRN Electronic Journal*, 2018, doi: 10.2139/ssrn.3258935.
- [7] T. Handhayani, J. Hendryli, and L. Hiryanto, “Comparison of shallow and deep learning models for classification of Lasem batik patterns,” in *Proceedings - 2017 1st International Conference on Informatics and Computational Sciences, ICICoS 2017*, 2017, vol. 2018-January. doi: 10.1109/ICICOS.2017.8276330.
- [8] I. M. A. Agastya and A. Setyanto, “Classification of Indonesian batik using deep learning techniques and data augmentation,” in *Proceedings - 2018 3rd International Conference*

- on *Information Technology, Information Systems and Electrical Engineering, ICITISEE 2018*, 2018. doi: 10.1109/ICITISEE.2018.8720990.
- [9] Y. Gultom, A. M. Arymurthy, and R. J. Masikome, "Batik Classification using Deep Convolutional Network Transfer Learning," *Jurnal Ilmu Komputer dan Informasi*, vol. 11, no. 2, 2018, doi: 10.21609/jiki.v11i2.507.
- [10] A. Y. Wicaksono, N. Suciati, C. Fatichah, K. Uchimura, and G. Koutaki, "Modified Convolutional Neural Network Architecture for Batik Motif Image Classification," *IPTEK Journal of Science*, vol. 2, no. 2, 2017, doi: 10.12962/j23378530.v2i2.a2846.
- [11] B. Ghogh and M. Crowley, "The Theory Behind Overfitting, Cross Validation, Regularization, Bagging, and Boosting: Tutorial," May 2019.
- [12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun ACM*, vol. 60, no. 6, 2017, doi: 10.1145/3065386.
- [13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, vol. 2016-December. doi: 10.1109/CVPR.2016.90.
- [14] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," in *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, 2020. doi: 10.1609/aaai.v34i07.7000.
- [15] P. Chen, S. Liu, H. Zhao, and J. Jia, "GridMask Data Augmentation," Jan. 2020.
- [16] S. Zagoruyko and N. Komodakis, "Wide Residual Networks," in *British Machine Vision Conference 2016, BMVC 2016*, 2016, vol. 2016-September. doi: 10.5244/C.30.87.
- [17] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017, vol. 2017-January. doi: 10.1109/CVPR.2017.243.
- [18] D. Han, J. Kim, and J. Kim, "Deep pyramidal residual networks," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017, vol. 2017-January. doi: 10.1109/CVPR.2017.668.
- [19] "Google Images." <https://images.google.com/>