

Draft Pengajuan HaKi

Penentuan Epoch pada Arsitektur Convolutional Neural Network  
Klasifikasi Citra Daun Tanaman Padi



Di Ajukan Oleh :

Fauzan Masykur, ST, M.Kom  
Mohammad Bhanu Setyawan, ST, M.Kom  
Kuntang Winangun S.Pd M.Pd  
Ellisia Kumalasari, S.Pd, M.Pd

UNIVERSITAS MUHAMMADIYAH PONOROGO  
Juli, 2022

## Ringkasan

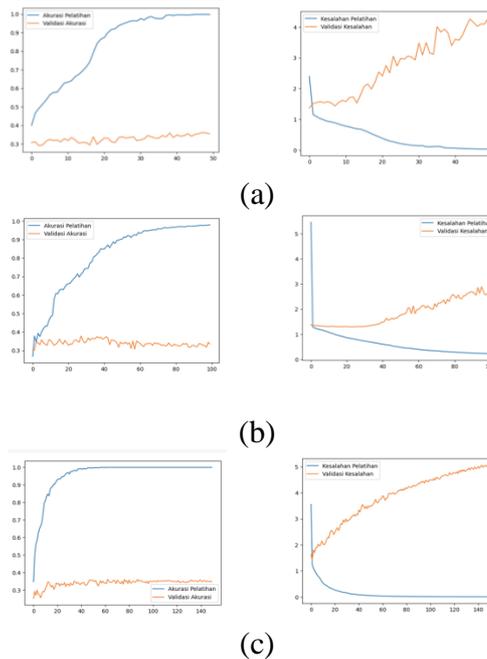
*Epoch* pada proses klasifikasi citra menggunakan algoritma *deep learning convolutional neural network* (CNN) merupakan proses yang mutlak dibutuhkan untuk mencapai akurasi paling optimal dan *loss* yang kecil. Penentuan *epoch* akan mempengaruhi bobot (*weight*) pada *neural network* serta berpengaruh pada kurva yang dihasilkan. Proses training dataset citra pada *neural network* akan melalui satu rangkaian perhitungan dari awal sampai akhir hingga akan dikembalikan lagi ke proses awal dengan sebutan sekali putaran atau 1 *epoch*. Proses pembelajaran dari dataset diperlukan beberapa kali *update* bobot sehingga akan kurang optimal jika *epoch* dilakukan hanya sekali putaran. Pada makalah ini akan dipaparkan klasifikasi citra daun tanaman padi yang menggunakan arsitektur CNN *MobileNet* dengan beberapa kali *epoch* untuk menghasilkan hasil yang paling optimal. Citra daun dibagi menjadi 4 kelas yakni *healthy*, *brownspot*, *hispa* dan *leafblast*. Setiap kelas di kelompokkan menjadi 2 kelompok yaitu data training dan data validasi dengan presentase data *training* 70% dan data *testing* 30%. Sedangkan *epoch* yang ditentukan sebanyak 50, 100 dan 150 kali. Pada *epoch* 150 menghasilkan nilai akurasi tertinggi senilai 1,000 dan *loss* senilai 0,0037 yang artinya semakin besar nilai *epoch* akan menghasilkan akurasi terbaik pada klasifikasi citra daun padi menggunakan *MobileNet*.

## Deskripsi

Pengaruh besarnya nilai *epoch* pada klasifikasi citra daun tanaman padi pada arsitektur *MobileNet* menjadi bahan pokok penelitian ini. Dataset diambil dari kaggle.com pada tahun 2019 melibatkan 4 kelas sudah melalui proses anotasi sehingga tinggal menggunakan pada google colab. Arsitektur *MobileNet* membagi konvolusi layer menjadi 2 bagian yakni *depthwise* dan *pointwise* konvolusi.

Rectifier Linier Unit (ReLU) sebagai fungsi akitifasi untuk menghilangkan nilai-nilai negatif pada hasil pooling. Proses pooling menggunakan maxpooling untuk mengurangi nilai pixel citra besar namun tidak mengurangi ciri dari citra tersebut. Pengulangan pooling dan konvolusi membuat citra menjadi lebih dikenali dan mampu mengurangi overfitting yang sering terjadi pada citra ukuran besar. Grafik 1 hasil klasifikasi dengan 3 nilai Epoch yang berbeda (50,100,150) menghasilkan grafik yang berbeda. Pada epoch 50 (grafik 1a) menunjukkan grafik akurasi belum stabil pada angka 1 yang artinya tingkat akurasi belum bisa dijadikan acuan sebagai nilai akhir klasifikasi. Demikian pada bagian

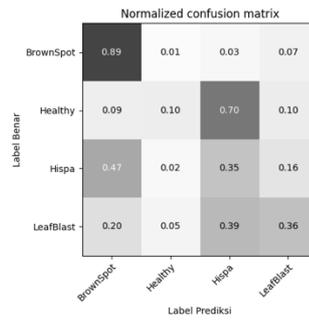
grafik loss dimana kesalahan akurasi dari awal epoch sampai epoch 50 menunjukkan tren penurunan yang artinya loss semakin berkurang namun belum menunjukkan nilai semestinya. Grafik 1b dan 1c menunjukkan pergerakan nilai akurasi yang semakin smooth dari epoch 1 hingga epoch 150 yang artinya epoch 150 menjadi lebih baik nilai akurasinya demikian juga dengan nilai loss. Akurasi mencapai nilai 1 saat epoch pada posisi 62 hingga epoch 150 dan nilai loss terkecil senilai 0,037 pada epoch 150.



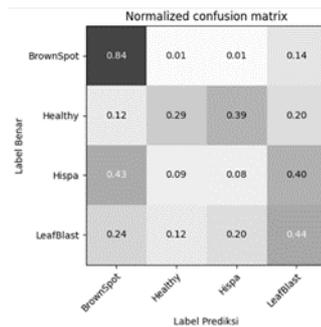
**Grafik 1.** Akurasi dan Kesalahan Pada Setiap Epoch

(a) *Epoch 50* (b) *Epoch 100* (c) *Epoch 150*

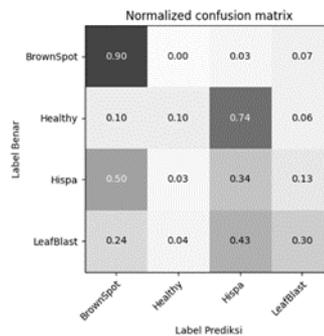
Sementara itu normalisasi dengan matrik konfusi menghasilkan perbandingan label prediksi dengan label aktual seperti pada gambar 1,2,3. Normalisasi ketiga nilai epoch menunjukkan adanya nilai lebih baik pada Epoch 150. Kelas Brownsport pada masing-masing epoch secara berurutan dari Epoch 50, 100 dan 150 menunjukkan nilai 0.89, 0.84 dan 0.90. Dengan demikian nilai epoch 150 menunjukkan hasil prediksi lebih baik dengan selisih 0.1



**Gambar 1.** Normalisasi pada Epoch 50

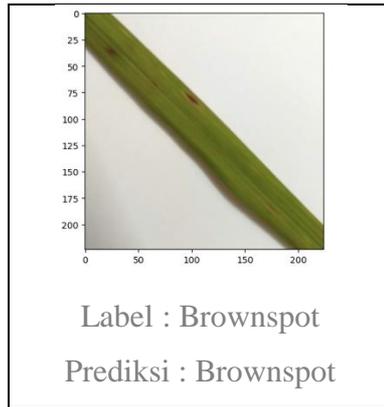


**Gambar 2.** Normalisasi pada Epoch 100



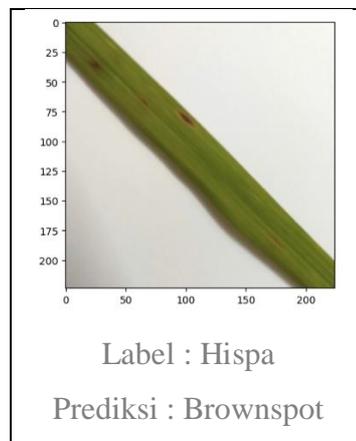
**Gambar 3.** Normalisasi pada Epoch 100

Proses epoch menunjukkan perlunya proses training dalam 1 putaran hingga di ulang-ulang untuk mendapatkan nilai yang terbaik. Proses training tidaklah cukup jika hanya dilakukan sekali pada kumpulan dataset dengan jumlah ribuan karena itu perlu adanya sebuah looping training. Gambar 4 menunjukkan hasil prediksi dari hasil proses training pada epoch 50.



**Gambar 4.** Hasil Prediksi Sesuai Dengan Label

Prediksi tidak selalu sama dengan kelas yang ditentukan, adakalanya prediksi juga salah tidak sesuai dengan kelas. Gambar 5 menunjukkan kesalahan prediksi pada Epoch 150. Ketidaktepatan prediksi tidaklah menunjukkan adanya ketidaksesuaian arsitektur dan ketidaktepatan prediksi masih dalam tingkat kewajaran seperti ditunjukkan pada grafik 1.



**Gambar 5.** Kesalahan Prediksi Pada Epoch 150

## Proses Running Source Code

```
Epoch 1/50
10/10 [=====] - 552s 54s/step - loss: 2.3983 - acc: 0.4003 - val_loss: 1.3706 - val_acc: 0.3074
Epoch 2/50
10/10 [=====] - 170s 17s/step - loss: 1.1579 - acc: 0.4654 - val_loss: 1.5140 - val_acc: 0.3108
Epoch 3/50
10/10 [=====] - 162s 17s/step - loss: 1.0870 - acc: 0.4916 - val_loss: 1.5449 - val_acc: 0.2905
Epoch 4/50
10/10 [=====] - 160s 16s/step - loss: 1.0245 - acc: 0.5135 - val_loss: 1.5756 - val_acc: 0.2939
Epoch 5/50
10/10 [=====] - 159s 16s/step - loss: 0.9799 - acc: 0.5363 - val_loss: 1.5393 - val_acc: 0.3142
Epoch 6/50
10/10 [=====] - 157s 16s/step - loss: 0.9370 - acc: 0.5642 - val_loss: 1.5750 - val_acc: 0.3243
Epoch 7/50
10/10 [=====] - 158s 16s/step - loss: 0.9168 - acc: 0.5777 - val_loss: 1.5324 - val_acc: 0.3176
Epoch 8/50
10/10 [=====] - 158s 16s/step - loss: 0.8869 - acc: 0.5785 - val_loss: 1.4347 - val_acc: 0.3209
Epoch 9/50
10/10 [=====] - 159s 17s/step - loss: 0.8472 - acc: 0.6030 - val_loss: 1.5481 - val_acc: 0.3108
Epoch 10/50
10/10 [=====] - 159s 16s/step - loss: 0.8096 - acc: 0.6284 - val_loss: 1.6154 - val_acc: 0.3277
Epoch 11/50
10/10 [=====] - 158s 16s/step - loss: 0.7778 - acc: 0.6326 - val_loss: 1.5791 - val_acc: 0.3176
Epoch 12/50
```