Klasifikasi Variasi Beras Menggunakan Deep Learning dengan VGG16

ANGGOTA :
MUHAMMAD HUSNI 22416255201212
INA AMELIA 2241&255201293

Pendahuluan

Penggunaan teknologi dalam pertanian telah berkembang pesat, terutama dalam klasifikasi jenis beras. Salah satu teknologi yang sangat berpotensi adalah computer vision, yang memungkinkan analisis otomatis terhadap gambar-gambar biji beras untuk mengidentifikasi varietas beras dengan akurasi tinggi. Dalam konteks ini, kami mengembangkan sebuah sistem berbasis computer vision untuk mengklasifikasikan jenis beras menggunakan pendekatan deep learning.

Computer vision

Computer vision adalah bidang AI yang memungkinkan komputer dan sistem untuk mengambil informasi bermakna dari gambar digital, video, dan dokumen visual lainnya.

VGG16

kepanjangan dari Visual Geometric Group yang dikembangkan oleh Oxford University dan VGG16 memiliki arti yaitu terdapat 16layer yang terdapat dalam Visual Geometric Group [22]. Secara singkat VGG16 adalah salah satu konfigurasi arsitektur dalam Convolutional Neural Network.

DATA KASUS

Mengembangkan model deep learning yang mampu mengklasifikasikan varietas beras berdasarkan gambar-gambar biji beras. Ada tiga jenis beras, yaitu :

- Varietas Beras Basmathi
- Varietas Beras IR64
- Varietas Beras Ketan

TAHAPAN RENYELESAIAN

MASALAH

1.Pengumpulan Data gambar.
2.Preprocessing Data gambar.
3. Membangun dan melatih model CNN.
4. Pelatihan model .
5. Evaluasi dan Validasi.

IMPLEMENTASI RROGRAM

- Implementasi program ini dilakukan dengan menggunakan colab dan library TensorFlow serta Keras.
- Model CNN dibangun dan dilatih menggunakan data gambar.

Selanjutnya adalah Demo Program!

DATA

Basmathi

IR64

Ketan

```
test  train
Basmathi  IR64  Ketan
B10.jpg  B17.jpg  B23.jpg  B2.jpg   B36.jpg  B42.jpg  B49.jpg  B55.jpg  B61.jpg  B68.jpg  B74.jpg
B11.jpg  B18.jpg  B24.jpg  B30.jpg  B37.jpg  B43.jpg  B4.jpg   B56.jpg  B62.jpg  B69.jpg  B75.jpg
B12.jpg  B19.jpg  B25.jpg  B31.jpg  B38.jpg  B44.jpg  B50.jpg  B57.jpg  B63.jpg  B6.jpg   B7.jpg
B13.jpg  B1.jpg   B26.jpg  B32.jpg  B39.jpg  B45.jpg  B51.jpg  B58.jpg  B64.jpg  B70.jpg  B8.jpg
B14.jpg  B20.jpg  B27.jpg  B33.jpg  B3.jpg   B46.jpg  B52.jpg  B59.jpg  B65.jpg  B71.jpg  B9.jpg
B15.jpg  B21.jpg  B28.jpg  B34.jpg  B40.jpg  B47.jpg  B53.jpg  B5.jpg   B66.jpg  B72.jpg
B16.jpg  B22.jpg  B29.jpg  B35.jpg  B41.jpg  B48.jpg  B54.jpg  B60.jpg  B67.jpg  B73.jpg
I10.jpg  I17.jpg  I23.jpg  I2.jpg   I36.jpg  I42.jpg  I49.jpg  I55.jpg  I61.jpg  I68.jpg  I74.jpg
I11.jpg  I18.jpg  I24.jpg  I30.jpg  I37.jpg  I43.jpg  I4.jpg   I56.jpg  I62.jpg  I69.jpg  I75.jpg
I12.jpg  I19.jpg  I25.jpg  I31.jpg  I38.jpg  I44.jpg  I50.jpg  I57.jpg  I63.jpg  I6.jpg   I7.jpg
I13.jpg  I1.jpg   I26.jpg  I32.jpg  I39.jpg  I45.jpg  I51.jpg  I58.jpg  I64.jpg  I70.jpg  I8.jpg
I14.jpg  I20.jpg  I27.jpg  I33.jpg  I3.jpg   I46.jpg  I52.jpg  I59.jpg  I65.jpg  I71.jpg  I9.jpg
I15.jpg  I21.jpg  I28.jpg  I34.jpg  I40.jpg  I47.jpg  I53.jpg  I5.jpg   I66.jpg  I72.jpg
I16.jpg  I22.jpg  I29.jpg  I35.jpg  I41.jpg  I48.jpg  I54.jpg  I60.jpg  I67.jpg  I73.jpg
K10.jpg  K17.jpg  K23.jpg  K2.jpg   K36.jpg  K42.jpg  K49.jpg  K55.jpg  K61.jpg  K68.jpg  K74.jpg
K11.jpg  K18.jpg  K24.jpg  K30.jpg  K37.jpg  K43.jpg  K4.jpg   K56.jpg  K62.jpg  K69.jpg  K75.jpg
K12.jpg  K19.jpg  K25.jpg  K31.jpg  K38.jpg  K44.jpg  K50.jpg  K57.jpg  K63.jpg  K6.jpg   K7.jpg
K13.jpg  K1.jpg   K26.jpg  K32.jpg  K39.jpg  K45.jpg  K51.jpg  K58.jpg  K64.jpg  K70.jpg  K8.jpg
K14.jpg  K20.jpg  K27.jpg  K33.jpg  K3.jpg   K46.jpg  K52.jpg  K59.jpg  K65.jpg  K71.jpg  K9.jpg
K15.jpg  K21.jpg  K28.jpg  K34.jpg  K40.jpg  K47.jpg  K53.jpg  K5.jpg   K66.jpg  K72.jpg
K16.jpg  K22.jpg  K29.jpg  K35.jpg  K41.jpg  K48.jpg  K54.jpg  K60.jpg  K67.jpg  K73.jpg
```

FEATURE EXTRACTION

Menggunakan model *pre-trained* untuk ekstraksi fitur (*feature extraction*) : Ketika bekerja dengan dataset kecil, adalah umum untuk mengambil keuntungan dari fitur yang dipelajari oleh model yang dilatih pada dataset yang lebih besar dalam domain yang sama. Ini dilakukan dengan membuat contoh model *pre-trained* dan menambahkan classifier yang sepenuhnya terhubung di atas

```
Model: "vgg16"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 224, 224, 3)]     0
 block1_conv1 (Conv2D)       (None, 224, 224, 64)      1792
 block1_conv2 (Conv2D)       (None, 224, 224, 64)      36928
 block1_pool (MaxPooling2D)  (None, 112, 112, 64)      0
 block2_conv1 (Conv2D)       (None, 112, 112, 128)     73856
 block2_conv2 (Conv2D)       (None, 112, 112, 128)     147584
 block2_pool (MaxPooling2D)  (None, 56, 56, 128)       0
 block3_conv1 (Conv2D)       (None, 56, 56, 256)       295168
 block3_conv2 (Conv2D)       (None, 56, 56, 256)       590080
 block3_conv3 (Conv2D)       (None, 56, 56, 256)       590080
```

```
 block4_conv2 (Conv2D)       (None, 28, 28, 512)       2359808
 block4_conv3 (Conv2D)       (None, 28, 28, 512)       2359808
 block4_pool (MaxPooling2D)  (None, 14, 14, 512)       0
 block5_conv1 (Conv2D)       (None, 14, 14, 512)       2359808
 block5_conv2 (Conv2D)       (None, 14, 14, 512)       2359808
 block5_conv3 (Conv2D)       (None, 14, 14, 512)       2359808
 block5_pool (MaxPooling2D)  (None, 7, 7, 512)         0
=================================================================
Total params: 14714688 (56.13 MB)
Trainable params: 0 (0.00 Byte)
Non-trainable params: 14714688 (56.13 MB)
_________________________________________________________________
```

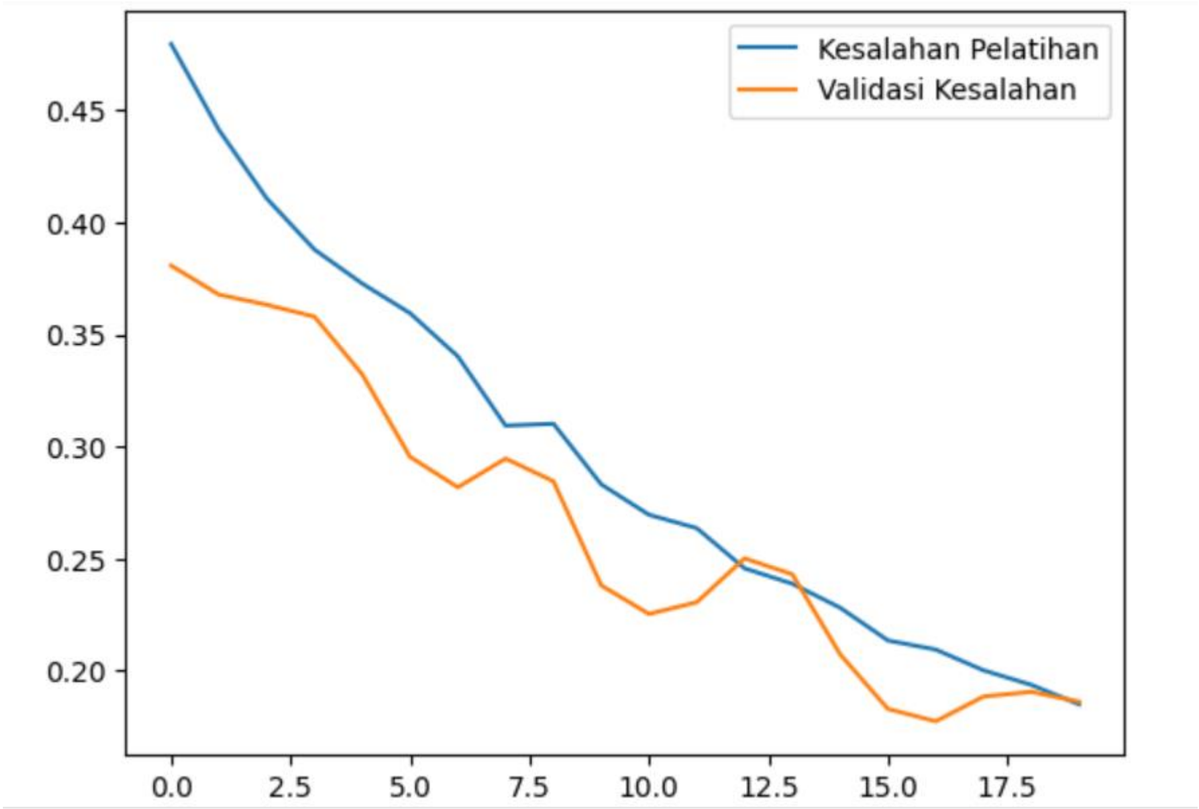
Pembuatan Model

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 vgg16 (Functional)          (None, 7, 7, 512)         14714688

 conv2d (Conv2D)             (None, 5, 5, 32)          147488

 global_average_pooling2d (  (None, 32)                0
 GlobalAveragePooling2D)

 dense (Dense)               (None, 3)                 99

=================================================================
Total params: 14862275 (56.70 MB)
Trainable params: 147587 (576.51 KB)
Non-trainable params: 14714688 (56.13 MB)
_________________________________________________________________
```

Pelatihan

```
<ipython-input-20-60b97efbd098>:5: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version.
  history = model.fit_generator(train_generator,
Epoch 1/20
2/2 [==============================] - 3s 859ms/step - loss: 0.4795 - acc: 0.8722 - val_loss: 0.3807 - val_acc: 0.9778
Epoch 2/20
2/2 [==============================] - 3s 3s/step - loss: 0.4412 - acc: 0.8833 - val_loss: 0.3678 - val_acc: 0.9556
Epoch 3/20
2/2 [==============================] - 3s 2s/step - loss: 0.4107 - acc: 0.8889 - val_loss: 0.3633 - val_acc: 0.9333
Epoch 4/20
2/2 [==============================] - 3s 2s/step - loss: 0.3879 - acc: 0.8778 - val_loss: 0.3579 - val_acc: 0.8444
Epoch 5/20
2/2 [==============================] - 3s 1s/step - loss: 0.3727 - acc: 0.8944 - val_loss: 0.3322 - val_acc: 0.8889
Epoch 6/20
2/2 [==============================] - 3s 2s/step - loss: 0.3595 - acc: 0.9167 - val_loss: 0.2954 - val_acc: 0.9333
Epoch 7/20
2/2 [==============================] - 3s 903ms/step - loss: 0.3402 - acc: 0.9167 - val_loss: 0.2819 - val_acc: 0.9333
Epoch 8/20
2/2 [==============================] - 3s 3s/step - loss: 0.3093 - acc: 0.9444 - val_loss: 0.2947 - val_acc: 0.9111
Epoch 9/20
2/2 [==============================] - 3s 2s/step - loss: 0.3102 - acc: 0.9167 - val_loss: 0.2846 - val_acc: 0.8889
Epoch 10/20
2/2 [==============================] - 3s 2s/step - loss: 0.2832 - acc: 0.9556 - val_loss: 0.2381 - val_acc: 0.9333
Epoch 11/20
```

Hasil pelatihan

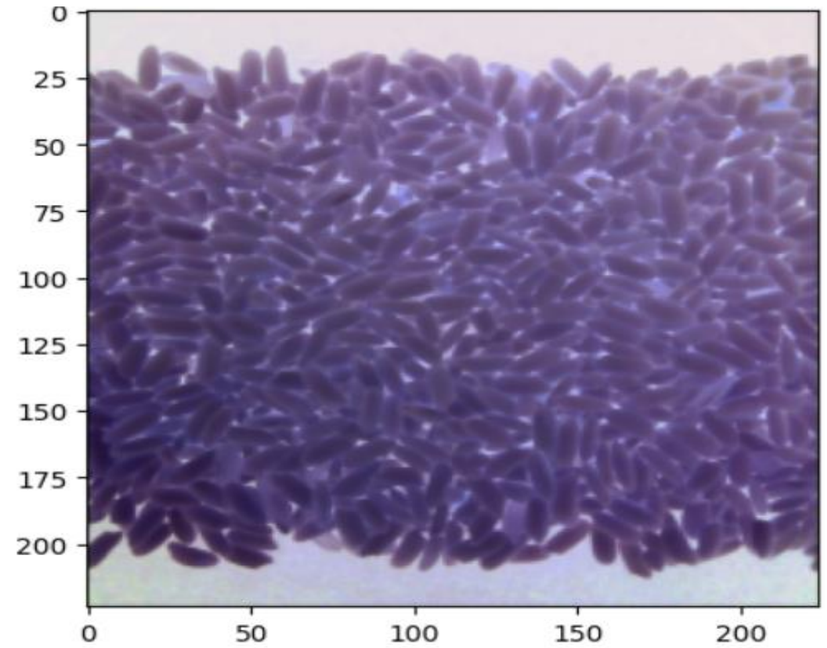


Menggunakan Model

```
#Prediksi Label Validasi dengan Pelatihan
n = 44
input_image = image_batch[n][np.newaxis,...]
print("Labelnya adalah: ", label_batch[n])

predictions = model.predict(input_image)
print("Prediksinya adalah",predictions[0])
```

Prediksi Gambar Individual

```
[ ]
```




```
Label yang benar adalah: 2 : Ketan
1/1 [==============================] - 0s 18ms/step
Nilai yang diprediksi adalah: [0.00857556 0.9877753  0.00364917]
Label yang diprediksi adalah: 1 : IR64
Prediksi salah
```

Kesimpulan

Projek ini berhasil mengembangkan sistem klasifikasi varietas beras menggunakan algoritma Convolutional Neural Network (CNN). Dengan menerapkan teknologi Computer Vision, sistem ini mampu mengidentifikasi dan mengklasifikasikan varietas beras dengan akurasi tinggi.

Thank YOU