

IMPLEMENTASI CONVOLUTIONAL NEURAL NETWORK MENGUNAKAN LIBRARY TENSORFLOW UNTUK DETEKSI KESEGARAN PADA APEL

Oleh:

Diana Cindy Agustin

Progam Studi Informatika

Universitas Muhammadiyah Sidoarjo

2023



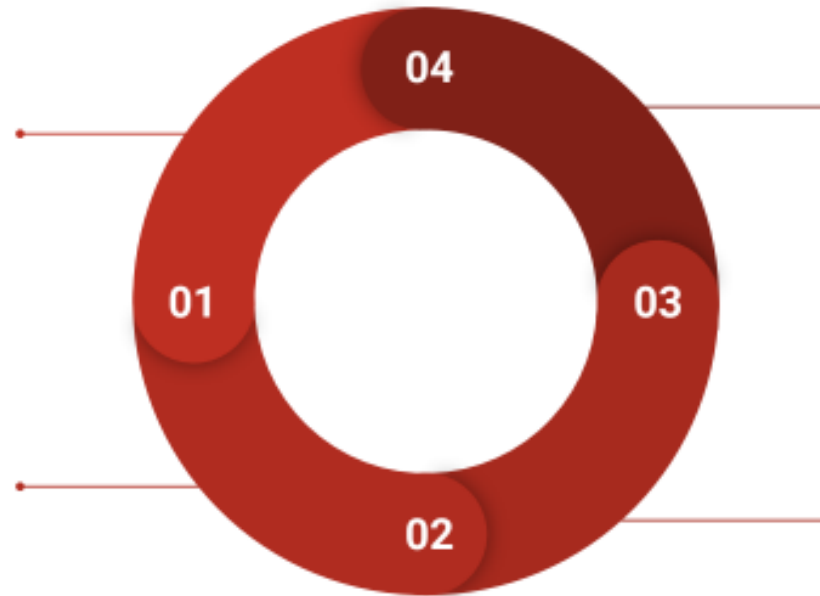
Alur Penelitian

Project Planning & Setup

Pada fase ini meliputi identifikasi masalah dan alokasi sumber daya

Data Collection & Labeling

Pada fase ini meliputi pengumpulan data, data preprocessing, dan augmentasi data



Deployment & Monitoring

Pada fase ini, meliputi memasukkan model ke dalam produksi perangkat lunak yang diperlukan untuk membuat model berjalan, dan membuat prediksi. Kita juga perlu memantau dan memelihara sistem.

Model Training & Debugging

Pada fase ini meliputi pembuatan model, training dan evaluasi model, membuat prediksi, dan test image

Data Collection dan Labeling

Langkah pertama dari penelitian ini adalah pengumpulan data yang diambil melalui Kaggle, setelah data dikumpulkan selanjutnya adalah melakukan data preprocessing dengan memisahkan data menjadi data training dan data test untuk memeriksa keakuratan model setelah training. Augmentasi data dilakukan untuk membantu mengatasi masalah umum pada deep learning yaitu data hungry (membutuhkan banyak data) juga data overfitting. Pada penelitian kali ini menggunakan augmentasi data flip, random crop, random rotation, dan shear sehingga menghasilkan 652 citra baru.

Bagian	Jumlah Data	Kelas	Jumlah
Data train	2611	<i>Fresh Apples</i>	1355
		<i>Rotten Apples</i>	1256
Data validation	652	<i>Fresh Apples</i>	338
		<i>Rotten Apples</i>	314
Data test	996	<i>Fresh Apples</i>	395
		<i>Rotten Apples</i>	601

Model Training dan Debugging

Model yang digunakan dalam penelitian ini terdiri dari convolutional layer, pooling layer, dropout layer, flatten layer, dan fully connected layer. Setelah model terbentuk kemudian akan dilakukan training citra menggunakan model fit dengan 15 kali iterasi atau epoch. Proses training dengan menunjukkan perubahan accuracy dan loss

```
Epoch 1/15
/usr/local/lib/python3.8/dist-packages/keras/backend.py:5585: UserWarning: ``sparse_categorical_crossentropy`` received `from_logits`
output, from_logits = _get_logits(
131/131 [=====] - 45s 245ms/step - loss: 1.1987 - accuracy: 0.8035 - val_loss: 3.7096 - val_accuracy: 0.67
Epoch 2/15
131/131 [=====] - 27s 210ms/step - loss: 0.3062 - accuracy: 0.8870 - val_loss: 1.8754 - val_accuracy: 0.86
Epoch 3/15
131/131 [=====] - 27s 208ms/step - loss: 0.2765 - accuracy: 0.8809 - val_loss: 1.4554 - val_accuracy: 0.85
Epoch 4/15
131/131 [=====] - 27s 210ms/step - loss: 0.2638 - accuracy: 0.8970 - val_loss: 0.9090 - val_accuracy: 0.88
Epoch 5/15
131/131 [=====] - 27s 209ms/step - loss: 0.2634 - accuracy: 0.8958 - val_loss: 0.6390 - val_accuracy: 0.96
Epoch 6/15
131/131 [=====] - 28s 211ms/step - loss: 0.2305 - accuracy: 0.9004 - val_loss: 0.5078 - val_accuracy: 0.91
Epoch 7/15
131/131 [=====] - 28s 211ms/step - loss: 0.2638 - accuracy: 0.9004 - val_loss: 0.5560 - val_accuracy: 0.91
Epoch 8/15
131/131 [=====] - 28s 211ms/step - loss: 0.2262 - accuracy: 0.9108 - val_loss: 1.0620 - val_accuracy: 0.85
Epoch 9/15
131/131 [=====] - 28s 211ms/step - loss: 0.2239 - accuracy: 0.9134 - val_loss: 0.3951 - val_accuracy: 0.85
Epoch 10/15
131/131 [=====] - 27s 209ms/step - loss: 0.2218 - accuracy: 0.9192 - val_loss: 1.2602 - val_accuracy: 0.87
Epoch 11/15
131/131 [=====] - 32s 243ms/step - loss: 0.2169 - accuracy: 0.9119 - val_loss: 0.4587 - val_accuracy: 0.85
Epoch 12/15
131/131 [=====] - 27s 208ms/step - loss: 0.2253 - accuracy: 0.9119 - val_loss: 6.1676 - val_accuracy: 0.66
Epoch 13/15
131/131 [=====] - 27s 206ms/step - loss: 0.2226 - accuracy: 0.9203 - val_loss: 0.2365 - val_accuracy: 0.92
Epoch 14/15
131/131 [=====] - 27s 208ms/step - loss: 0.2187 - accuracy: 0.9184 - val_loss: 0.5563 - val_accuracy: 0.87
Epoch 15/15
131/131 [=====] - 32s 243ms/step - loss: 0.2027 - accuracy: 0.9230 - val_loss: 0.3722 - val_accuracy: 0.96
```

Model Training dan Debugging

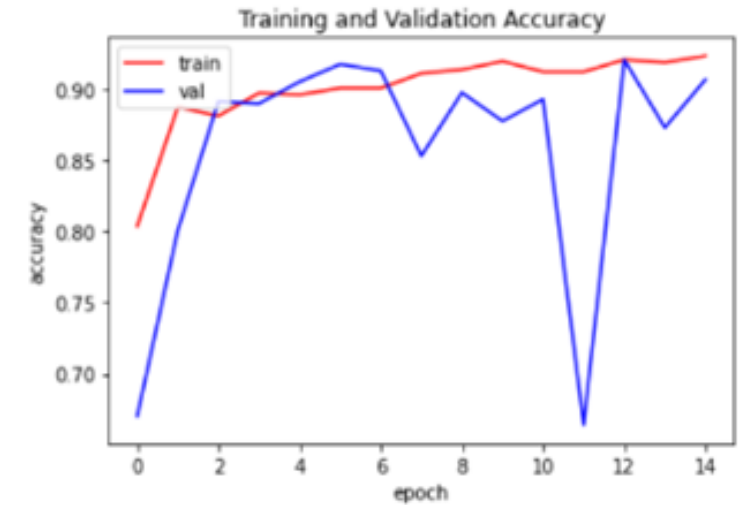
Setelah dilakukan training maka model akan melakukan evaluasi pada 996 data test untuk melihat tingkat akurasi dari model yang sudah dibuat dalam membaca dan mengklasifikasikan data

```
Found 996 images belonging to 2 classes.  
996/996 [=====] - 10s 10ms/step - loss: 0.1555 - accuracy: 0.9357  
accuracy test: 0.935742974281311
```

akurasi yang ditampilkan mencapai angka 0.93 atau 93% dari maksimal 100% tingkat akurasi yang bisa didapat oleh sebuah model

Model Training dan Debugging

Proses training yang dilakukan oleh model menunjukkan validation accuracy yang didapat relatif stabil pada iterasi 0 sampai iterasi 11 kemudian ada perubahan cukup signifikan di satu titik dari iterasi 11 ke 12 dan cenderung stabil kembali pada iterasi 12 sampai selesai. Validation loss juga menunjukkan hasil yang sama, ada perubahan signifikan pada satu titik saja yakni iterasi ke 12. Dapat dilihat bahwa semakin bertambahnya jumlah epoch maka semakin naik grafik accuracy dan semakin turun grafik loss yang didapat. Hal ini menjadi baik karena itu artinya validation loss semakin rendah dan tingkat validation accuracy semakin naik. Model dianggap sudah cukup baik karena memiliki akurasi tinggi dan loss rendah



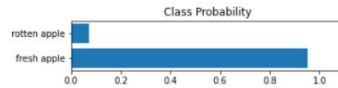
Model Training dan Debugging

Selanjutnya dilakukan tes model menggunakan data yang berbeda dari dataset train dan test, menunjukkan hasil yang sesuai dan model dapat mengklasifikasikan buah tersebut segar atau busuk berdasarkan kelasnya.

**Terdeteksi
“fresh apple”**

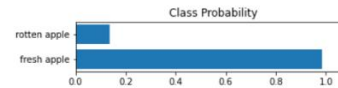
1/1 [=====] - 0s 22ms/step

fresh apple



1/1 [=====] - 0s 22ms/step

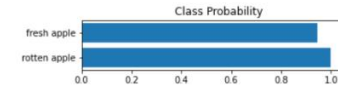
fresh apple



**Terdeteksi
“fresh apple”**

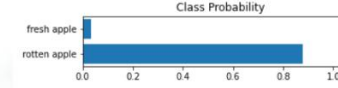
1/1 [=====] - 0s 24ms/step

rotten apple



1/1 [=====] - 0s 21ms/step

rotten apple



**Terdeteksi
“rotten apple”**

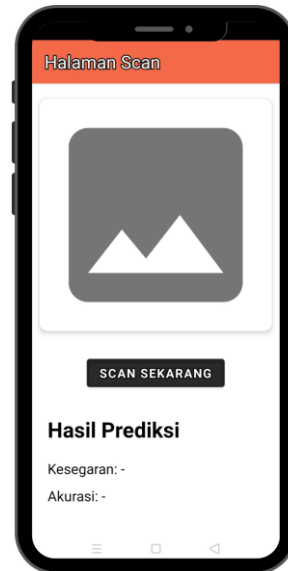
**Terdeteksi
“rotten apple”**

Deployment & Monitoring

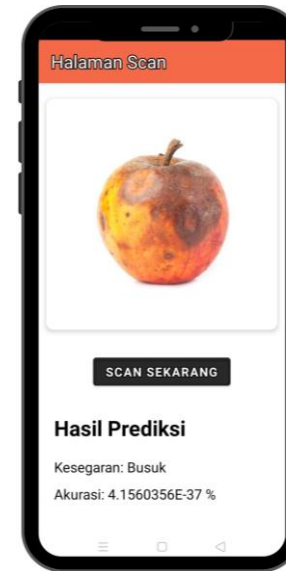
Pada penelitian kali ini model diekstrak menjadi format Flatbuffer (.tflite). Setelah diekstrak, model kemudian di-deploy ke dalam android untuk memudahkan visualisasi dari program yang sudah dibuat.



Splash screen



Input gambar



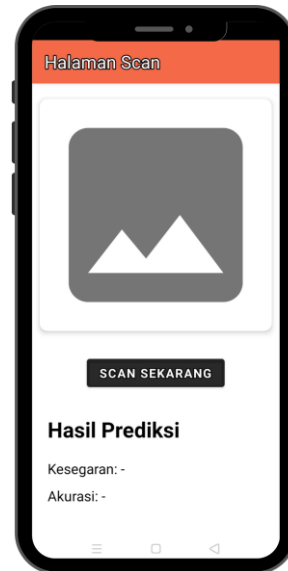
Hasil klasifikasi

Deployment & Monitoring

Pada penelitian kali ini model diekstrak menjadi format Flatbuffer (.tflite). Setelah diekstrak, model kemudian di-deploy ke dalam android untuk memudahkan visualisasi dari program yang sudah dibuat.



Splash screen

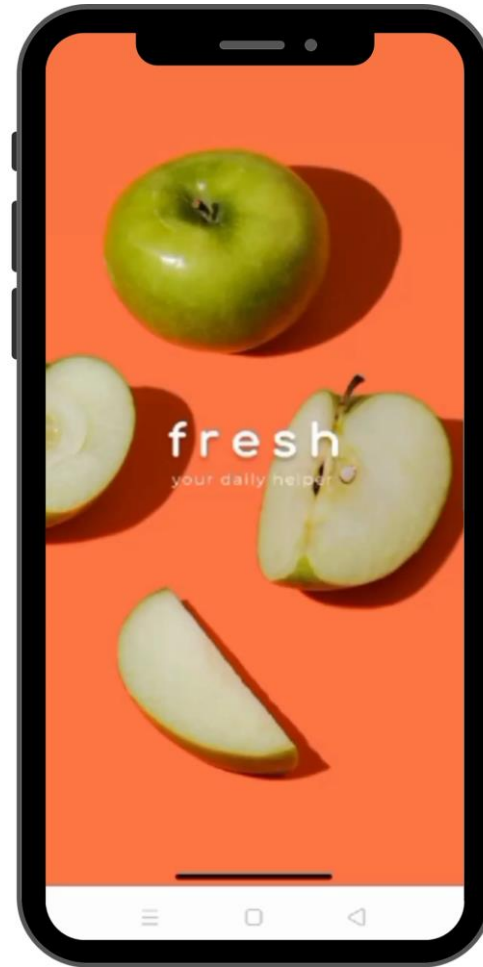


Input gambar



Hasil klasifikasi

Deployment & Monitoring



Referensi

- Adi Nugroho, P., Fenriana, I., & Arijanto, R. (2020). IMPLEMENTASI DEEP LEARNING MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK (CNN) PADA EKSPRESI MANUSIA. JURNAL ALGOR, 2(1). <https://jurnal.buddhidharma.ac.id/index.php/algor/index>
- Dou, Z., Toth, J. D., & Westendorff, F. (2020). DISEASE DETECTION SYSTEM IN CASSAVA LEAVES USING ANDROID-BASED DEEP LEARNING AND TENSORFLOW. In IJIS Indonesian Journal on Information System.
- f, M. L. (2018). Food waste for livestock feeding: Feasibility, safety, and sustainability implications. In Global Food Security (Vol. 17, pp. 154–161). Elsevier B.V. <https://doi.org/10.1016/j.gfs.2017.12.003>
- Faturrahman, M., & Yustiana, I. (n.d.). DISEASE DETECTION SYSTEM IN CASSAVA LEAVES USING ANDROID-BASED DEEP LEARNING AND TENSORFLOW. In IJIS Indonesian Journal on Information System.
- Homepage, J., Roihan, A., Abas Sunarya, P., & Rafika, A. S. (2019). IJCIT (Indonesian Journal on Computer and Information Technology) Pemanfaatan Machine Learning dalam Berbagai Bidang: Review paper. In IJCIT (Indonesian Journal on Computer and Information Technology) (Vol. 5, Issue 1).
- Ilahiyah, S., & Nilogiri, A. (n.d.). Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network.
- Jayasankar, K., Karthika, B., Jeyashree, T., Deepalakshmi, R., & Karthika, G. (2018). Fruit Freshness Detection Using Raspberry PI. International Journal of Innovative Research in Applied Sciences and Engineering, 1(10), 202. <https://doi.org/10.29027/ijirase.v1.i10.2018.202-208>
- Juliansyah, S., & Laksito, A. D. (2021). Klasifikasi Citra Buah Pir Menggunakan Convolutional Neural Networks. Jurnal Telekomunikasi dan Komputer, 11(1), 65. <https://doi.org/10.22441/incomtech.v11i1.10185>
- Kuriakose, A. B., Siju, F., Aji, M. K., Rahim, T., & Rani, C. (n.d.). Automatic Fruit Classification and Freshness Detection. www.ijert.org
- Liunanda, C. N., Rostianingsih, S., & Purbowo, A. N. (n.d.). Implementasi Algoritma YOLO pada Aplikasi Pendeteksi Senjata Tajam di Android.
- Lorentius, C. A., Adipranata, R., & Tjondrowiguno, A. (n.d.). Pengenalan Aksara Jawa dengan Menggunakan Metode Convolutional Neural Network.
- Mehindra Prasmatio, R., Rahmat, B., & Yuniar, I. (2020). ALGORITMA CONVOLUTIONAL NEURAL NETWORK. In Jurnal Informatika dan Sistem Informasi (JIFoSI) (Vol. 1, Issue 2).
- Nugroho, B., & Puspaningrum, E. Y. (2021). Kinerja Metode CNN untuk Klasifikasi Pneumonia dengan Variasi Ukuran Citra Input. Jurnal Teknologi Informasi dan Ilmu Komputer, 8(3), 533. <https://doi.org/10.25126/jtiik.2021834515>
- Paritosh, K., Kushwaha, S. K., Yadav, M., Pareek, N., Chawade, A., & Vivekanand, V. (2017). Food Waste to Energy: An Overview of Sustainable Approaches for Food Waste Management and Nutrient Recycling. In BioMed Research International (Vol. 2017). Hindawi Limited. <https://doi.org/10.1155/2017/2370927>
- Prastika, I. W., Zuliarso, E., Lomba, J. T., No, J., & 50241, S. (2021). DETEKSI PENYAKIT KULIT WAJAH MENGGUNAKAN TENSORFLOW DENGAN METODE CONVOLUTIONAL NEURAL NETWORK. Jurnal Manajemen Informatika & Sistem Informasi, 4(2). <http://e-journal.stmiklombok.ac.id/index.php/misi>
- Wulandari, I., Yasin, H., & Widiyari, T. (2020). Klasifikasi Citra Digital Bumbu Dan Rempah Dengan Algoritma Convolutional Neural Network (Cnn). Jurnal Gaussian, 9(3), 273–282. <https://doi.org/10.14710/j.gauss.v9i3.27416>
- Yuliany, S., & Nur Rachman, A. (2022). Implementasi Deep Learning pada Sistem Klasifikasi Hama Tanaman Padi Menggunakan Metode Convolutional Neural Network (CNN). Jurnal Buana Informatika, 13(1), 54–65.

