

SKRIPSI

**PERHITUNGAN JUMLAH DAN PREDIKSI TINGKAT
KEMATANGAN BUAH DARI CITRA POHON
BUAH-BUAHAN**



Edward Tjahyadi

NPM: 6181801064

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
2023**

UNDERGRADUATE THESIS

**PREDICTING FRUITS NUMBER AND THEIR RIPENESS
LEVEL FROM IMAGES OF FRUIT TREES**



Edward Tjahyadi

NPM: 6181801064

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
2023**

LEMBAR PENGESAHAN

**PERHITUNGAN JUMLAH DAN PREDIKSI TINGKAT
KEMATANGAN BUAH DARI CITRA POHON BUAH-BUAHAN**

Edward Tjahyadi

NPM: 6181801064

Bandung, 6 Juli 2023

Menyetujui,

Pembimbing

Dr. Veronica Sri Moertini

Ketua Tim Penguji

Anggota Tim Penguji

Keenan Adiwijaya Leman, M.T.

Maria Veronica, M.T.

Mengetahui,

Ketua Program Studi

Mariskha Tri Adithia, P.D.Eng

PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa skripsi dengan judul:

PERHITUNGAN JUMLAH DAN PREDIKSI TINGKAT KEMATANGAN BUAH DARI CITRA POHON BUAH-BUAHAN

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,
Tanggal 6 Juli 2023



Edward Tjahyadi
NPM: 6181801064

ABSTRAK

Salah satu masalah yang dihadapi dalam bidang perkebunan ini adalah faktor *human error* yang dapat mempengaruhi hasil panen. Faktor *human error* yang dimaksud adalah penentuan tingkat kematangan buah. Untuk dapat meminimalisir faktor *human error* dibuatlah penelitian yang bertujuan untuk membuat model yang dapat memprediksi tingkat kematangan suatu buah hanya dengan memasukkan input berupa gambar buah. Buah yang digunakan dalam penelitian ini terdiri dari 3 jenis buah yaitu jeruk, mangga dan pepaya.

Untuk dapat melatih model yang diinginkan, tahap awal yang dilakukan adalah melakukan eksperimen mengenai *deep learning* dan berbagai macam teknik yang dapat digunakan dalam pembuatan model untuk melakukan deteksi objek pada gambar. Pelatihan model dilakukan dengan menggunakan salah satu algoritma *deep learning* yaitu *Convolutional Neural Network* (CNN). Arsitektur CNN yang digunakan dalam pelatihan model adalah arsitektur RetinaNet. Arsitektur ini terdiri dari 2 komponen penting yaitu *feature extraction* dan prediksi. *Feature extraction* dilakukan dengan memanfaatkan *backbone Feature Pyramid Network* (FPN). FPN merupakan arsitektur *deep learning* yang terdiri dari lapisan-lapisan gambar dimana setiap lapisan memiliki ukuran gambar yang berbeda dengan fitur yang berbeda-beda. FPN diawali dengan *bottom-up pathway* yang bertugas untuk melakukan *downsampling* pada gambar *input* dengan tujuan mencari fitur pada setiap ukuran gambar, dengan semakin mengecilnya ukuran gambar maka fitur yang didapatkan akan bernilai sangat berharga. Tahap selanjutnya dilakukan *upsampling* pada *top-down pathway*, pada jalur ini akan dilakukan penggabungan fitur pada setiap tingkat piramida dan menghubungkan setiap lapisan piramida dengan *lateral connection*. Dari setiap lapisan piramida akan dihasilkan fitur-fitur yang dapat digunakan untuk memprediksi lokasi objek dan kelas objek, prediksi ini dilakukan dengan memanfaatkan *classification subnet* dan *box regression subnet*.

Pelatihan model yang dibuat menghasilkan model untuk memprediksi tingkat kematangan masing-masing jenis buah yang telah ditentukan (Jeruk, Mangga dan Pepaya) dengan model terbaik untuk buah jeruk mendapatkan nilai *mean Average Precision* 0.82 dan presisi sebesar 0.85 untuk memprediksi buah jeruk matang dan 0.79 untuk memprediksi buah jeruk tidak matang dengan *hyperparameter* yang digunakan adalah 2 layer yang digunakan untuk konvolusi pada *classification subnet* dan *regression subnet*, dengan 4 x 4 filter yang digunakan pada *classification subnet* dan *regression subnet* dan menggunakan 4 *pyramid level*. Untuk buah mangga mendapatkan nilai *mean Average Precision* 0.83 dan presisi sebesar 0.85 untuk buah mangga matang dan 0.82 untuk buah mangga tidak matang dengan *hyperparameter* yang digunakan adalah 4 layer yang digunakan untuk konvolusi pada *classification subnet* dan *regression subnet* dan dengan 3 x 3 filter yang digunakan pada *classification subnet* dan *regression subnet*. Sedangkan untuk buah pepaya model yang dihasilkan kurang baik dengan nilai presisi sebesar 0.71 untuk buah pepaya matang dan 0.33 untuk buah pepaya tidak matang.

Kesimpulan yang dapat diambil dari penelitian ini adalah model yang dihasilkan untuk memprediksi tingkat kematangan buah jeruk dan mangga sudah cukup baik menurut penulis namun untuk buah pepaya model yang dihasilkan kurang baik. Hal ini disebabkan karena gambar buah pepaya yang digunakan dalam pembuatan model saling berhimpitan satu sama lain.

Kata-kata kunci: Industri 4.0, Perkebunan, Buah, Tingkat Kematangan, *Deep Learning*, *Convolutional Neural Network*.

ABSTRACT

One of the problems faced in the field of agriculture is the factor of human error, which can affect the crop yield. The human error factor referred to here is the determination of fruit ripeness. To minimize human error, research is conducted with the aim of creating a model that can predict the level of fruit ripeness by simply inputting a fruit image. The fruits used in this research consist of three types: oranges, mangoes, and papayas.

To train the desired model, the initial stage involves experimenting with deep learning and various techniques that can be used to create a model for object detection in images. The model training is performed using a deep learning algorithm called Convolutional Neural Network (CNN). The CNN architecture used in the model training is RetinaNet. This architecture consists of two important components: feature extraction and prediction. Feature extraction is performed using the Feature Pyramid Network (FPN) as the backbone. FPN is a deep learning architecture consisting of image layers, where each layer has a different image size and different features. FPN starts with a bottom-up pathway that downsamples the input image to find features at each image size. As the image size decreases, the obtained features become more valuable. The next stage involves upsampling in the top-down pathway, where feature fusion is performed at each pyramid level and each pyramid layer is connected with lateral connections. From each pyramid layer, features are generated that can be used to predict the location and class of objects. This prediction is carried out using the classification subnet and box regression subnet.

The created model for predicting the ripeness level of each fruit type (Orange, Mango, and Papaya) achieved the best performance for ripe oranges with a mean Average Precision score of 0.82 and a precision score of 0.85 for predicting ripe oranges and 0.79 for predicting unripe oranges. The hyperparameters used for the model include 2 convolutional layers for both the classification subnet and regression subnet, using 4x4 filters, and 4 pyramid levels. For mangoes, the model achieved a mean Average Precision score of 0.83 and a precision score of 0.85 for ripe mangoes and 0.82 for unripe mangoes. The hyperparameters used for mangoes include 4 convolutional layers for both the classification subnet and regression subnet, using 3x3 filters. However, the model for papayas performed poorly with a precision score of 0.71 for ripe papayas and 0.33 for unripe papayas.

In conclusion, the model produced for predicting the ripeness level of oranges and mangoes is considered good according to the author's evaluation. However, the model for papayas performed poorly. This is attributed to the overlapping of papaya fruit images used in the model creation process.

Keywords: Industry 4.0, Plantation, Fruit, Ripeness level, *Deep Learning*, *Convolutional Neural Network*.

*Skripsi ini penulis persembahkan kepada pembimbing skripsi Dr.
Veronica Sri Moertini, keluarga dan teman-teman yang telah
membantu penulis selama pembuatan skripsi ini.*

KATA PENGANTAR

Puji, hormat dan syukur kepada Tuhan yang Maha Esa atas berkat yang telah diberikan, penulis dapat menyelesaikan skripsi dengan judul "Perhitungan Jumlah dan Prediksi Tingkat Kematangan Buah dari Citra Pohon Buah-Buahan". Skripsi ini disusun untuk memenuhi syarat kelulusan di program studi sarjana Informatika UNPAR.

Penulis menyadari bahwa skripsi yang ditulis masih belum sempurna dan masih mengalami kesulitan dan hambatan. Akan tetapi, masih banyak pihak yang telah membantu penulis dalam hal bimbingan, motivasi, dukungan dan doa. Penulis mengucapkan terima kasih kepada semua pihak yang telah membantu penulis dalam menyelesaikan skripsi ini. Secara khusus penulis mengucapkan terima kasih sebesar-besarnya kepada:

- Keluarga penulis yang telah memberikan dukungan dan doa selama pengerjaan skripsi ini.
- Ibu Dr. Veronica Sri Moertini selaku dosen pembimbing yang telah meluangkan waktu dan tenaga untuk mendampingi penulis dengan penuh kesabaran.
- Rama Fauzi, Ariq Pratama, Obed Kristiaji, Gian Martin, Shannas Rizqi, Josie, Wiranata, Syahdan, Yohanes Irfon, Ame Fedora dan seluruh teman saya yang tidak dapat saya sebutkan satu persatu yang membantu memberikan dorongan motivasi dan ilmu untuk penulisan skripsi ini.
- Kepada seluruh teman yang bukan berasal dari universitas yang sama yang memberikan dukungan motivasi.
- Pihak kebun yang penulis kunjungi yang telah meluangkan waktu selama tahap pengumpulan data.

Penulis menyadari bahwa skripsi ini buakn karya yang sempurna. Maka dari itu, penulis terbuka atas kritik dan saran dari pembaca yang dapat menyempurnakan penelitian ini. Semoga skripsi ini dapat bermanfaat bagi pembaca.

Bandung, Juli 2023

Penulis

DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxiii
DAFTAR KODE PROGRAM	xxv
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan	3
1.4 Batasan Masalah	3
1.5 Metodologi	3
1.6 Sistematika Pembahasan	3
2 LANDASAN TEORI	5
2.1 Buah	5
2.2 Citra	7
2.3 Data Science	8
2.3.1 Data	8
2.3.2 Tahapan Data Science	9
2.4 Deep Learning	9
2.4.1 <i>Convolutional Neural Network</i>	12
2.4.2 RetinaNet	15
2.4.3 Evaluasi model	17
3 ANALISIS PENYELESAIAN MASALAH	21
3.1 Analisis Masalah	21
3.2 Ide Solusi	21
3.3 LabelImg	21
3.4 Keras-RetinaNet	23
3.4.1 Tensorflow	23
3.4.2 Keras	24
3.4.3 PIL	24
3.4.4 NumPy	25
3.5 Flask	25
4 PENAMBANGAN DATA	27
4.1 Pengumpulan Data	27
4.2 Penyiapan Data	27
4.3 Eksperimen Tahap Awal	31

4.4	Pembuatan Model	36
4.4.1	Eksperimen <i>hyperparameter</i> Buah Jeruk	38
4.4.2	Jeruk	53
4.4.3	Eksperimen <i>hyperparameter</i> Buah Mangga	58
4.4.4	Mangga	72
4.4.5	Pepaya	75
5	PELUNCURAN MODEL DAN PENGUJIAN	79
5.1	Lingkungan Perangkat Keras	79
5.2	Lingkungan Perangkat Lunak	79
5.3	Pembuatan Perangkat Lunak	79
5.3.1	Use case dan Skenario	79
5.3.2	Data Flow Diagram	80
5.3.3	Pembuatan Mock-up	81
5.3.4	Implementasi Program	81
5.3.5	Pengujian Perangkat Lunak	83
6	KESIMPULAN DAN SARAN	93
6.1	Kesimpulan	93
6.2	Saran	94
	DAFTAR REFERENSI	95
	A KODE PROGRAM	97
	B HASIL EKSPERIMEN	119

DAFTAR GAMBAR

2.1	Contoh buah jeruk dekopon mentah.	5
2.2	Contoh buah jeruk dekopon matang.	5
2.3	Contoh buah pepaya taiwan mentah.	6
2.4	Contoh buah pepaya taiwan matang.	6
2.5	Contoh buah mangga mentah.	6
2.6	Contoh buah mangga matang.	6
2.7	Matrix <i>RGB</i> [1].	7
2.8	<i>Color space HSV</i> [2].	7
2.9	Matrix <i>Grayscale</i> [3].	8
2.10	Matrix Biner[4].	8
2.11	<i>Biological Neural Network</i> [5].	9
2.12	<i>Artificial neuron</i> [6].	10
2.13	Fungsi aktivasi [7].	11
2.14	Arsitektur CNN[8].	12
2.15	<i>Convolutional operation</i> [9].	12
2.16	Mendemonstrasikan langkah dari <i>max pooling</i> dengan ukuran <i>pooling</i> sebesar 2 x 2 [10].	13
2.17	Sigmoid	14
2.18	Tanh	14
2.19	ReLU[10].	15
2.20	<i>Leaky ReLU</i> [10].	15
2.21	<i>Fully Connected (FC) Layer</i> [10].	15
2.22	Arsitektur <i>RetinaNet</i> [11].	16
2.23	Arsitektur dari <i>Feature Pyramid Network</i> [12].	16
2.24	Focal Loss[11].	18
2.25	<i>Confusion matrix</i>	19
2.26	Cara penilaian <i>IoU</i>	19
2.27	Contoh nilai <i>IoU False Positive</i> dan <i>True Positive</i>	20
2.28	Contoh nilai <i>IoU</i>	20
3.1	Pelabelan dilakukan pada objek yang ingin dideteksi	22
4.1	Contoh data gambar jeruk yang diambil.	27
4.2	Contoh data gambar mangga yang diambil.	28
4.3	Contoh data gambar pepaya yang diambil.	28
4.4	Ilustrasi pelabelan dengan menggunakan <i>tool labelImg</i>	28
4.5	Contoh data gambar yang digunakan untuk eksperimen awal.	31
4.6	<i>Average Precision</i> global	32
4.7	<i>Average Precision</i> untuk Jeruk Matang	33
4.8	<i>Average Precision</i> untuk Jeruk Tidak Matang	33
4.9	<i>Average Precision</i> untuk Jeruk Matang(atas) dan <i>Average Precision</i> untuk Jeruk Tidak Matang(bawah)	34
4.10	<i>Average Precision</i> Global	35

4.11	<i>Average Precision</i> untuk Jeruk Matang	35
4.12	<i>Average Precision</i> untuk Jeruk Tidak Matang	36
4.13	Hasil Debug.	37
4.14	<i>Average Precision</i> untuk Jeruk Matang(atas) dan Jeruk Tidak Matang(bawah). . .	39
4.15	Nilai <i>loss</i> pada eksperimen dengan menggunakan 6 layer.	39
4.16	<i>Average Precision</i> untuk Jeruk Matang(atas) dan Jeruk Tidak Matang(bawah). . .	40
4.17	Nilai <i>loss</i> pada eksperimen dengan menggunakan 2 layer.	40
4.18	<i>Average Precision</i> untuk Jeruk Matang(atas) dan Jeruk Tidak Matang(bawah). . .	41
4.19	Nilai <i>loss</i> pada eksperimen dengan menggunakan 1 layer.	41
4.20	<i>Average Precision</i> untuk Jeruk Matang(atas) dan Jeruk Tidak Matang(bawah). . .	42
4.21	Nilai <i>loss</i> pada eksperimen dengan menggunakan 1 layer dan ukuran <i>filter</i> 4 x 4. .	42
4.22	<i>Average Precision</i> untuk Jeruk Matang(atas) dan Jeruk Tidak Matang(bawah). . .	43
4.23	Nilai <i>loss</i> pada eksperimen dengan menggunakan 2 layer dan ukuran <i>filter</i> 3 x 3. .	43
4.24	<i>Average Precision</i> untuk Jeruk Matang(atas) dan Jeruk Tidak Matang(bawah). . .	44
4.25	Nilai <i>loss</i> pada eksperimen dengan menggunakan 2 layer dan ukuran <i>filter</i> 2 x 2. .	44
4.26	<i>Average Precision</i> untuk Jeruk Matang(atas) dan Jeruk Tidak Matang(bawah). . .	45
4.27	Nilai <i>loss</i> pada eksperimen dengan menggunakan 2 layer dan ukuran <i>filter</i> 4 x 4. .	45
4.28	<i>Average Precision</i> untuk Jeruk Matang(atas) dan Jeruk Tidak Matang(bawah). . .	46
4.29	Nilai <i>loss</i> pada eksperimen dengan menggunakan 2 layer dan ukuran <i>filter</i> 5 x 5. .	46
4.30	<i>Average Precision</i> untuk Jeruk Matang(atas) dan Jeruk Tidak Matang(bawah). . .	47
4.31	Nilai <i>loss</i> pada eksperimen dengan menggunakan 2 layer, ukuran <i>filter</i> 4 x 4 dan <i>pyramid level</i> menjadi 3.	47
4.32	<i>Average Precision</i> untuk Jeruk Matang(atas) dan Jeruk Tidak Matang(bawah). . .	48
4.33	Nilai <i>loss</i> pada eksperimen dengan menggunakan 2 layer, ukuran <i>filter</i> 4 x 4 dan <i>pyramid level</i> menjadi 4.	48
4.34	<i>Average Precision</i> untuk Jeruk Matang(atas) dan Jeruk Tidak Matang(bawah). . .	49
4.35	Nilai <i>loss</i> pada eksperimen dengan menggunakan 2 layer, ukuran <i>filter</i> 4 x 4 dan <i>pyramid level</i> menjadi 5.	49
4.36	<i>Average Precision</i> untuk Jeruk Matang(atas) dan Jeruk Tidak Matang(bawah). . .	50
4.37	Nilai <i>loss</i> pada eksperimen dengan menggunakan 2 layer, ukuran <i>filter</i> 4 x 4 dan <i>pyramid level</i> menjadi 6.	50
4.38	<i>Average Precision</i> untuk Jeruk Matang(atas) dan Jeruk Tidak Matang(bawah). . .	51
4.39	Nilai <i>loss</i> pada eksperimen dengan menggunakan 2 layer, ukuran <i>filter</i> 4 x 4 dan <i>pyramid level</i> menjadi 7.	51
4.40	<i>Average Precision</i> untuk Jeruk Matang(atas) dan Jeruk Tidak Matang(bawah). . .	52
4.41	Nilai <i>loss</i> pada eksperimen dengan menggunakan 8 layer dan ukuran <i>filter</i> 8 x 8. .	52
4.42	Contoh gambar jeruk sederhana.	54
4.43	<i>Average Precision</i> untuk Jeruk Matang(atas) dan Jeruk Tidak Matang(bawah). . .	54
4.44	<i>Average Precision</i> untuk Jeruk Matang(atas) dan Jeruk Tidak Matang(bawah). . .	55
4.45	Contoh gambar jeruk yang digunakan pada pelatihan model kedua.	56
4.46	<i>Average Precision</i> untuk Jeruk Matang(atas) dan Jeruk Tidak Matang(bawah). . .	56
4.47	<i>Average Precision</i> untuk Jeruk Matang(atas) dan Jeruk Tidak Matang(bawah). . .	57
4.48	<i>Average Precision</i> untuk Jeruk Matang(atas) dan Jeruk Tidak Matang(bawah). . .	58
4.49	<i>Average Precision</i> untuk Mangga Matang(atas) dan Mangga Tidak Matang(bawah). .	59
4.50	Nilai <i>loss</i> pada eksperimen dengan menggunakan 1 layer dan ukuran <i>filter</i> 2 x 2. .	60
4.51	<i>Average Precision</i> untuk Jeruk Matang(atas) dan Jeruk Tidak Matang(bawah). . .	60
4.52	Nilai <i>loss</i> pada eksperimen dengan menggunakan 1 layer dan ukuran <i>filter</i> 3 x 3. .	60
4.53	<i>Average Precision</i> untuk Jeruk Matang(atas) dan Jeruk Tidak Matang(bawah). . .	61
4.54	Nilai <i>loss</i> pada eksperimen dengan menggunakan 1 layer dan ukuran <i>filter</i> 5 x 5. .	61
4.55	<i>Average Precision</i> untuk Jeruk Matang(atas) dan Jeruk Tidak Matang(bawah). . .	62
4.56	Nilai <i>loss</i> pada eksperimen dengan menggunakan 2 layer dan ukuran <i>filter</i> 4 x 4. .	62

4.57	<i>Average Precision</i> untuk Jeruk Matang(atas) dan Jeruk Tidak Matang(bawah).	63
4.58	Nilai <i>loss</i> pada eksperimen dengan menggunakan 2 layer dan ukuran filter 2 x 2.	63
4.59	<i>Average Precision</i> untuk Jeruk Matang(atas) dan Jeruk Tidak Matang(bawah).	64
4.60	Nilai <i>loss</i> pada eksperimen dengan menggunakan 3 layer dan ukuran filter 2 x 2.	64
4.61	<i>Average Precision</i> untuk Jeruk Matang(atas) dan Jeruk Tidak Matang(bawah).	65
4.62	Nilai <i>loss</i> pada eksperimen dengan menggunakan 4 layer dan ukuran filter 2 x 2.	65
4.63	<i>Average Precision</i> untuk Jeruk Matang(atas) dan Jeruk Tidak Matang(bawah).	66
4.64	Nilai <i>loss</i> pada eksperimen dengan menggunakan 5 layer dan ukuran filter 2 x 2.	66
4.65	<i>Average Precision</i> untuk Jeruk Matang(atas) dan Jeruk Tidak Matang(bawah).	67
4.66	Nilai <i>loss</i> pada eksperimen dengan menggunakan 1 layer, ukuran filter 2 x 2 dan <i>pyramid level</i> menjadi 3.	67
4.67	<i>Average Precision</i> untuk Jeruk Matang(atas) dan Jeruk Tidak Matang(bawah).	68
4.68	Nilai <i>loss</i> pada eksperimen dengan menggunakan 1 layer, ukuran filter 2 x 2 dan <i>pyramid level</i> menjadi 4.	68
4.69	<i>Average Precision</i> untuk Jeruk Matang(atas) dan Jeruk Tidak Matang(bawah).	69
4.70	Nilai <i>loss</i> pada eksperimen dengan menggunakan 1 layer, ukuran filter 2 x 2 dan <i>pyramid level</i> menjadi 5.	69
4.71	<i>Average Precision</i> untuk Jeruk Matang(atas) dan Jeruk Tidak Matang(bawah).	70
4.72	Nilai <i>loss</i> pada eksperimen dengan menggunakan 1 layer, ukuran filter 2 x 2 dan <i>pyramid level</i> menjadi 6.	70
4.73	<i>Average Precision</i> untuk Jeruk Matang(atas) dan Jeruk Tidak Matang(bawah).	71
4.74	Nilai <i>loss</i> pada eksperimen dengan menggunakan 1 layer, ukuran filter 2 x 2 dan <i>pyramid level</i> menjadi 7.	71
4.75	<i>Average Precision</i> untuk Jeruk Matang(atas) dan Jeruk Tidak Matang(bawah).	72
4.76	Nilai <i>loss</i> pada eksperimen dengan menggunakan 1 layer dan ukuran filter 4 x 4.	72
4.77	Contoh gambar mangga yang digunakan.	74
4.78	<i>Average Precision</i> untuk Mangga Matang(atas) dan Mangga Tidak Matang(bawah).	74
4.79	<i>Average Precision</i> untuk Mangga Matang(atas) dan Mangga Tidak Matang(bawah).	75
4.80	<i>Average Precision</i> untuk Mangga Matang(atas) dan Mangga Tidak Matang(bawah).	76
4.81	Contoh gambar pepaya yang digunakan.	76
4.82	<i>Average Precision</i> untuk Pepaya Matang(atas) dan Pepaya Tidak Matang(bawah).	77
5.1	<i>Use case</i> dari perangkat lunak yang dibuat.	80
5.2	<i>DFD level 0</i> dari perangkat lunak yang dibuat.	81
5.3	<i>DFD level 1</i> dari perangkat lunak yang dibuat.	81
5.4	Desain Mock-up	82
5.5	Tampilan awal perangkat lunak.	83
5.6	Tampilan setelah pengguna memasukkan input yang diinginkan.	84
5.7	Tampilan setelah pengguna memasukkan input yang diinginkan.	84
5.8	Contoh kasus jeruk 1.	85
5.9	Contoh kasus jeruk 2.	86
5.10	Contoh kasus jeruk 3.	87
5.11	Contoh kasus jeruk 4.	87
5.12	Contoh kasus jeruk 5.	88
5.13	Contoh kasus mangga 1.	88
5.14	Contoh kasus mangga 2	89
5.15	Contoh kasus mangga 3.	89
5.16	Contoh kasus 1.	90
5.17	Contoh kasus 2.	91
5.18	Contoh kasus 3.	91
B.1	<i>Average Precision</i> global	119

B.2	<i>Average Precision</i> untuk Jeruk Matang	119
B.3	<i>Average Precision</i> untuk Jeruk Tidak Matang	120
B.4	<i>Average Precision</i> untuk Jeruk Matang(atas) dan <i>Average Precision</i> untuk Jeruk Tidak Matang(bawah)	120
B.5	<i>Average Precision</i> Global	121
B.6	<i>Average Precision</i> untuk Jeruk Matang	121
B.7	<i>Average Precision</i> untuk Jeruk Tidak Matang	121
B.8	<i>Average Precision</i> untuk Jeruk Matang(atas) dan Jeruk Tidak Matang(bawah). . .	122
B.9	<i>Average Precision</i> untuk Jeruk Matang(atas) dan Jeruk Tidak Matang(bawah). . .	122
B.10	<i>Average Precision</i> untuk Jeruk Matang(atas) dan Jeruk Tidak Matang(bawah). . .	123
B.11	<i>Average Precision</i> untuk Jeruk Matang(atas) dan Jeruk Tidak Matang(bawah). . .	123
B.12	<i>Average Precision</i> untuk Jeruk Matang(atas) dan Jeruk Tidak Matang(bawah). . .	124
B.13	<i>Average Precision</i> untuk Mangga Matang(atas) dan Mangga Tidak Matang(bawah). . .	124
B.14	<i>Average Precision</i> untuk Mangga Matang(atas) dan Mangga Tidak Matang(bawah). . .	125
B.15	<i>Average Precision</i> untuk Mangga Matang(atas) dan Mangga Tidak Matang(bawah). . .	125
B.16	<i>Average Precision</i> untuk Pepaya Matang(atas) dan Pepaya Tidak Matang(bawah). . .	126

DAFTAR TABEL

2.1 Perbandingan <i>biological neural network</i> dengan <i>Artificial neural network</i>	10
4.1 Tabel jumlah gambar untuk setiap jenis buahnya dan jumlah setiap labelnya. . . .	29
4.2 Tabel jumlah masing-masing label pada setiap gambar untuk jenis buah jeruk. . .	30
4.3 Tabel jumlah masing-masing label pada setiap gambar untuk jenis buah pepaya. .	30
4.4 Tabel jumlah masing-masing label pada setiap gambar untuk jenis buah mangga. .	31
4.5 Parameter yang tidak diubah.	38
4.6 Parameter yang akan diubah.	38
4.7 Tabel perbandingan performa untuk setiap <i>hyperparameter</i>	53
4.8 Parameter yang akan diubah.	53
4.9 Parameter yang akan diubah.	58
4.10 Tabel perbandingan performa untuk setiap <i>hyperparameter</i>	73
4.11 Parameter yang akan diubah.	73
5.1 Tabel jumlah masing-masing label yang terdeteksi pada gambar untuk jenis buah jeruk.	84
5.2 Tabel jumlah masing-masing label yang terdeteksi pada gambar untuk jenis buah jeruk.	85
5.3 Tabel jumlah masing-masing label yang terdeteksi pada gambar untuk jenis buah jeruk.	85
5.4 Tabel jumlah masing-masing label yang terdeteksi pada gambar untuk jenis buah jeruk.	85
5.5 Tabel jumlah masing-masing label yang terdeteksi pada gambar untuk jenis buah jeruk.	86
5.6 Tabel jumlah masing-masing label yang terdeteksi pada gambar untuk jenis buah mangga.	86
5.7 Tabel jumlah masing-masing label yang terdeteksi pada gambar untuk jenis buah mangga.	87
5.8 Tabel jumlah masing-masing label yang terdeteksi pada gambar untuk jenis buah mangga.	89
5.9 Tabel jumlah masing-masing label yang terdeteksi pada gambar untuk jenis buah pepaya.	90
5.10 Tabel jumlah masing-masing label yang terdeteksi pada gambar untuk jenis buah pepaya.	90
5.11 Tabel jumlah masing-masing label yang terdeteksi pada gambar untuk jenis buah pepaya.	90

DAFTAR KODE PROGRAM

3.1	Command instalasi tool <code>labelImg</code>	22
3.2	Command untuk mengakses tool <code>labelImg</code>	22
3.3	Isi file xml hasil pelabelan.	22
3.4	Command instalasi kode program keras-retinanet	23
3.5	Kode program pemanggilan tensorflow dan numpy	23
3.6	Kode program pembuatan variabel baru.	23
3.7	Kode program pembuatan operasi komputasi yang diinginkan.	24
3.8	Kode program pembuatan operasi komputasi perkalian X1 dan X2.	24
3.9	Command instalasi Keras.	24
3.10	Command instalasi Python Imaging Library(PIL).	24
3.11	import Python Imaging Library(PIL).	24
3.12	import Python Imaging Library(PIL).	24
3.13	Command instalasi NumPy.	25
3.14	Command instalasi NumPy.	25
3.15	Command instalasi Flask.	25
3.16	Command instalasi Flask.	26
3.17	Command instalasi Flask.	26
4.1	Command untuk mengakses <i>tool</i> <code>labelImg</code>	28
4.2	Command Pembagian ratio data test dan train	32
4.3	Command Pembuatan Model	36
4.4	Command Pembagian ratio data <i>test</i> dan <i>train</i>	36
4.5	Command pelatihan model	37
A.1	<code>batch_rename.py</code>	97
A.2	<code>make_dataset.py</code>	97
A.3	<code>debug.py</code>	98
A.4	<code>pascal_voc.py</code>	101
A.5	<code>train.py</code>	103
A.6	<code>index.html</code>	109
A.7	<code>styles.css</code>	111
A.8	<code>script.js</code>	114
A.9	<code>detect_image.py</code>	116
A.10	<code>main.py</code>	117

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Revolusi Industri 4.0 yang berbasis pada jaringan internet sangat mempengaruhi semua sektor industri termasuk salah satunya adalah industri pertanian dan perkebunan. Salah satu penerapan Industri 4.0 pada bidang pertanian dan perkebunan adalah *smart farming*[13]. *Smart farming* ini sendiri bermanfaat untuk meningkatkan produktivitas, daya saing, dan keuntungan dalam pertanian dan perkebunan secara berkelanjutan. Pada *smart farming* terdapat sistem-sistem yang perlu dikembangkan untuk mengotomatisasi kegiatan pertanian dan perkebunan yang dapat membantu para petani untuk menjalankan tugasnya. Otomatisasi pada sistem pertanian diharapkan dapat meningkatkan efektivitas dalam bertani dan juga dapat meningkatkan kualitas hasil panen[14], salah satu sistem yang dapat diotomatisasi adalah pendeteksian jumlah buah dan tingkat kematangan buah pada pohon-pohon yang ditanam.

Pada dasarnya manusia dalam melakukan pekerjaannya memiliki kemungkinan untuk melakukan kesalahan, terlebih jika pekerjaan tersebut mengandalkan indera yang mereka miliki, dengan adanya faktor *human error*[15] tersebut maka terdapat kemungkinan perhitungan jumlah buah pada suatu pohon dan pengecekan tingkat kematangan buah menjadi kurang efektif karena akan memakan waktu yang lama dan juga memiliki tingkat akurasi yang rendah. Berdasarkan masalah tersebut perlu dilakukan penerapan metode baru untuk mendapatkan tingkat akurasi yang lebih tinggi. Salah satu metode baru yang bisa dikembangkan adalah penggunaan robot untuk membantu masalah panen. Robot tersebut nantinya akan memiliki kemampuan untuk menghitung jumlah buah dan menentukan tingkat kematangan buah, untuk itu robot perlu dilatih dengan model-model *deep learning*. Dengan diterapkannya metode baru ini, diharapkan masalah pertanian mengenai perhitungan buah dan penentuan tingkat kematangan dapat diatasi, sehingga dengan semakin efektifnya perhitungan buah, maka kuantitas hasil panen akan jauh lebih meningkat dan kualitas hasil panen juga akan meningkat. Dalam pelaksanaan penelitian ini dilakukan pengolahan citra pada buah dengan menggunakan teknik klasifikasi *deep learning*, dengan tujuan untuk mengatasi masalah penentuan jumlah buah dan juga tingkat kematangan buahnya. Dengan memanfaatkan komputer, harapannya komputer dapat mempelajari dan menduplikasi kemampuan manusia dalam memahami informasi yang terdapat pada citra [16].

Deep learning adalah sebuah bidang keilmuan dalam bidang *machine learning* dimana *deep learning* memiliki kemampuan yang sangat baik dalam kasus klasifikasi objek pada citra[16]. Cara kerja algoritma *Convolutional Neural Network* (CNN) diawali dengan *feature extraction* menggunakan *convolutional layer* dan *pooling layer*. Di dalam *convolutional layer* dibuatlah sebuah *filter* dengan ukuran yang dapat disesuaikan, dengan dibuatnya *filter* ini maka gambar yang ingin diolah akan menjadi lebih kecil secara dimensi. Langkah selanjutnya adalah *pooling layer*, tahap ini sama seperti yang dilakukan pada *convolutional layer*, dengan perbedaan *filter* yang dilakukan hanya mengambil nilai rata-rata atau maksimum, dengan demikian fitur kecil dalam beberapa piksel yang penting akan dipertahankan. Tahap selanjutnya adalah *fully-connected layer* setelah berkurangnya ukuran dimensi gambar, selanjutnya akan dilakukan klasifikasi[17].

Unit dasar dari beberapa algoritma *deep learning* adalah *neuron*. Konsep *neuron* didasarkan

pada *neuron* manusia. *Neuron* memiliki fungsi statistik yang menghitung rata-rata berdasarkan input dan menerapkan fungsi aktivasi ke hasil yang dihasilkan. *Layer* memiliki *neuron*, dengan setiap *layer* memiliki fungsi tertentu seperti *input layer* yang digunakan sebagai *input* untuk *layer* didepannya, lalu ada *hidden layer* yang memproses *input* dari *layer* sebelumnya, jumlah banyaknya *hidden layer* dapat diatur, semakin banyak jumlah *hidden layer* maka semakin dalam suatu neural network, lalu *layer* terakhir adalah *output layer* yang mengembalikan hasil prediksi dari informasi yang telah diolah[18]. Sistem CNN mungkin memiliki antara 3 hingga 150 atau bahkan lebih *layer*. *Layer CNN* dapat terdiri dari empat jenis utama: *Convolution Layer*, *ReLU Layer*, *Pooling Layer*, dan *Fully-Connected Layer*.

1. *Convolutional Layer*: Konvolusi adalah aplikasi sederhana dari *filter* ke *input* yang menghasilkan aktivasi. *Layer* konvolusi memiliki seperangkat *filter* yang dapat dilatih yang memiliki rentang penerimaan kecil tetapi dapat digunakan untuk data lengkap yang disediakan. *Layer* konvolusi adalah blok bangunan utama yang digunakan dalam jaringan saraf konvolusi.
2. *ReLU Layer*: *ReLU layer*, juga dikenal sebagai *Rectified linear unit layer*, adalah fungsi aktivasi yang diterapkan untuk menurunkan *overfitting* dan membangun akurasi dan efektivitas CNN. Model yang memiliki *layer* ini lebih mudah dilatih dan menghasilkan hasil yang lebih akurat.
3. *Pooling Layer*: *Layer* ini mengumpulkan hasil dari semua neuron di *layer* sebelumnya dan memproses data ini. Tugas utama dari *layer* penyatuan adalah untuk menurunkan jumlah faktor yang dipertimbangkan dan memberikan keluaran yang efisien.
4. *Fully-Connected Layer*: *Layer* ini adalah *layer* keluaran akhir untuk model CNN yang meratakan *input* yang diterima dari *layer* sebelumnya dan memberikan hasilnya.

Salah satu *arsitektur* yang dapat digunakan adalah *RetinaNet*. Untuk konstruksinya *RetinaNet* menggunakan *Feature Pyramid Network* (FPN). FPN sendiri merupakan jaringan saraf *convolutional* yang dibagun dalam bentuk piramida yang berfungsi untuk menggabungkan fitur pada setiap tingkatan piramida yang dimana gambar yang terletak di puncak piramida menghasilkan fitur yang paling berharga. Komponen-komponen penting dalam FPN terdiri dari *bottom-up pathway*, *top-down pathway* dan *lateral connection*. Pada *bottom-up pathway* dilakukan *downsampling* pada gambar *input* dengan tujuan untuk mendapatkan fitur yang secara semantik lebih tinggi dibandingkan dengan gambar yang memiliki dimensi yang lebih besar, *downsampling* ini dilakukan dengan menggunakan *ResNet*, dari setiap *level* piramida akan dihubungkan dengan *lateral connection*, tahap selanjutnya akan dilakukan *upsampling* pada *top-down pathway*, pada tahap ini juga akan dilakukan penggabungan fitur dari setiap lapisan pada piramida untuk memperkaya fitur yang akan digunakan.

ResNet sendiri merupakan *arsitektur* CNN yang mengandalkan *skip connection* dan *residual block*, penggunaan *ResNet* ini dapat mengurangi kemungkinan terjadinya *vanishing gradient* yang dapat menyebabkan tidak terjadinya perubahan nilai bobot dan nilai bias. Tidak terjadinya penyesuaian nilai bobot dan nilai bias dapat membuat model tidak berkembang.

TensorFlow merupakan salah satu *library open source* yang memiliki kinerja komputasi dan *machine learning* yang tinggi. *TensorFlow* bekerja dengan menggunakan *multidimensional array* yang disebut *tensor*. *Tensor* ini memungkinkan *Python* untuk menyelesaikan perhitungan yang lebih rumit yang diperlukan saat bekerja dengan *machine learning*. Format penyimpanan informasi ini digunakan untuk menyimpan lebih banyak lagi informasi rumit dalam *tensor* daripada dalam *array* satu dimensi.

Pada penelitian yang dilakukan oleh Wulan Anggraini[19] dan Tiara Shafira[20] dapat diketahui bahwa penggunaan algoritma *Convolution Neural Network (CNN)* pada deteksi citra memiliki tingkat akurasi yang tinggi, yaitu pada angka 90%. Hal inilah yang mendasari penelitian ini menggunakan metode *CNN* karena mampu membaca pengolahan citra dengan tingkat akurasi yang tinggi dalam mendeteksi objek maupun *image classification*, sehingga pada penelitian ini dikembangkan metode klasifikasi tingkat kematangan buah dan juga jumlah buahnya dengan menggunakan algoritma CNN.

1.2 Rumusan Masalah

Berikut ini adalah rumusan masalah dari latar belakang yang telah dipaparkan. Berdasarkan permasalahan di atas, berikut adalah permasalahan yang akan dikaji dalam penelitian ini :

1. Bagaimana cara menyiapkan data berupa citra buah dari setiap jenis buah yang telah ditentukan?
2. Bagaimana cara mendeteksi tingkat kematangan buah dari setiap jenis buah yang telah ditentukan?
3. Bagaimana cara mencari *hyperparameter* yang dapat menghasilkan model cukup baik?
4. Bagaimana cara mengevaluasi hasil deteksi tingkat kematangan dan perhitungan jumlah buah dari setiap jenis buah yang telah ditentukan?
5. Bagaimana cara membangun perangkat lunak untuk mendeteksi tingkat kematangan dan menghitung jumlah buah yang telah ditentukan?

1.3 Tujuan

Adapun tujuan penelitian dari rumusan masalah yang telah dipaparkan adalah:

1. Menyiapkan data berupa citra dari jenis buah yang telah ditentukan.
2. Melakukan deteksi objek pada gambar dengan memanfaatkan model yang dihasilkan.
3. Melakukan perubahan pada *hyperparameter* yang digunakan dalam pelatihan model.
4. Mengevaluasi model yang didapatkan terhadap hasil deteksi objek.
5. Membuat perangkat lunak untuk mendeteksi objek buah dan tingkat kematangan buah.

1.4 Batasan Masalah

Batasan masalah dalam penelitian ini adalah, perangkat lunak yang dibuat hanya dapat digunakan untuk jenis buah yang telah ditentukan.

1.5 Metodologi

Metodologi yang dilakukan pada penelitian ini adalah sebagai berikut:

1. Studi literatur mengenai *deep learning*.
2. Studi literatur mengenai *Tensorflow*.
3. Studi literatur mengenai tingkat kematangan buah.
4. Studi literatur mengenai citra.
5. Studi literatur mengenai *RetinaNet*.
6. Studi literatur mengenai algoritma *CNN*.
7. Mengumpulkan dan mengolah gambar buah yang telah ditentukan.
8. Melakukan pelabelan.
9. Melakukan pengujian pada data yang sudah dikumpulkan.
10. Melakukan eksperimen pembuatan model.
11. Menguji model yang sudah dibuat.
12. Membuat perangkat lunak.
13. Menguji model pada perangkat lunak yang dibuat.

1.6 Sistematika Pembahasan

Pembahasan penelitian dilakukan secara sistematis dengan detail sebagai berikut:

1. Bab 1 Pendahuluan
Berisi latar belakang dibuatnya skripsi Pendeteksian Tingkat Kematangan Buah dan Menghitung Jumlah Buahnya beserta rumusan masalah, batasan masalah, tujuan, metodologi dan sistematika pembahasan.
2. Bab 2 Landasan Teori
Berisi landasan teori yang mendukung penelitian ini seperti *deep learning*, *RetinaNet*, *Tensorflow*, buah dan *Convolutional Neural Network*.
3. Bab 3 Eksplorasi Perangkat Lunak
Berisi pembahasan mengenai hasil pembelajaran terhadap *tools-tools* yang digunakan dalam penelitian ini seperti *Tensorflow*, *Keras*, *Flask*, *PIL* dan *LabelImg*.
4. Bab 4 Analisis Masalah dan Eksperimen Tahap Awal
Berisi detail mengenai analisis masalah beserta ide solusi dari masalah tersebut, tahap pengumpulan data dan eksperimen yang dilakukan pada tahap awal yang berguna untuk menentukan *hyperparameter* yang akan digunakan untuk eksperimen selanjutnya.
5. Bab 5 Pembuatan dan Peluncuran Model
Berisi detail mengenai eksperimen yang dilakukan untuk pembuatan model untuk semua buah yang telah ditentukan, pembahasan mengenai kode program yang digunakan untuk melakukan pelatihan model beserta hasil analisis dari model yang dihasilkan dan penyebabnya dan pembuatan perangkat lunak beserta pengujiannya serta hasil analisis dari perangkat lunak yang telah dibuat.
6. Bab 6 Kesimpulan dan Saran
Berisi kesimpulan dan saran yang didapatkan dari hasil eksperimen yang telah dilakukan selama penelitian ini berlangsung.