

SISTEM PENGIRAAN UDANG
MENDASARKAN KAEDAH PEMBELAJARAN MENDALAM
DENGAN MENGGUNAKAN *MASK R-CNN*

TEH HONG KHAI

TESIS YANG DIKEMUKAKAN UNTUK MEMPEROLEHI
IJAZAH SARJANA MUDA SAINS KOMPUTER DENGAN KEPUJIAN

FAKULTI TEKNOLOGI DAN SAINS MAKLUMAT
UNIVERSITI KEBANGSAAN MALAYSIA

BANGI

2020

PENGAKUAN

Saya mengaku bahawa karya ini merupakan karya saya sendiri kecuali nukilan dan ringkasan yang telah saya jelaskan sumbernya.

18 Jun 2020



TEH HONG KHAI

A 166075

PENGHARGAAN

Terlebih dahulu saya ingin merakamkan setinggi-tinggi penghargaan dan jutaan terima kasih kepada penyelia saya, Assoc. Prof. Dr. Siti Norul Huda Sheikh Abdullah yang telah memberikan bimbingan, nasihat dan ajaran dengan penuh kesabaran. Sokongan dan nasihat positif yang turut disalurkan oleh beliau kepada saya juga telah membantu saya untuk mengharungi kemusykilan yang dihadapi sepanjang pelaksanaan usulan projek ini. Tidak dilupakan juga ketua-ketua program Sains Komputer, Dr. Suhaila Zainudin dan Dr. Nur Fazidah Elias yang sentiasa memberikan dorongan moral dan maklumat-maklumat terkini kepada semua mahasiswa Sains Komputer. Terima kasih juga disampaikan kepada Pengusaha Manjung Aquabest Hatchery iaitu Encik Ahmad Tarmizi yang telah banyak memberikan kerjasama dan ilmu kepada saya dalam menyiapkan penyediaan pangkalan data dalam projek ini.

Seterusnya, saya ingin menunjukkan penghargaan kepada ibu bapa dan ahli keluarga saya yang sentiasa memberikan sokongan moral serta mental. Jutaan terima kasih disampaikan kepada keluarga saya atas curahan kasih sayang yang tidak terhingga kepada saya. Di samping itu, saya ingin mengucapkan terima kasih kepada rakan-rakan yang telah memberikan sokongan moral dalam usaha saya melaksanakan projek ini. Sokongan mereka menaikkan keyakinan dan semangat saya untuk terus menempuh cabaran yang tertimbul sepanjang pelaksanaan projek ini.

Akhir sekali, ribuan terima kasih disampaikan kepada semua pihak yang terlibat secara langsung dan tidak langsung yang telah membantu saya dalam menyiapkan projek ini.

ABSTRAK

Pengiraan bermaksud perbuatan mengira sesuatu objek. Memang tidak boleh dinafikan bahawa manusia mampu mengira dengan senang. Walau bagaimanapun, manusia tidak mampu mengira secara berkesan jika jumlah objek yang terlalu banyak terdapat dalam imej. Hal ini disebabkan objek yang semakin banyak akan meningkatkan ketumpatan dan kerumitan imej serta menyusahkan proses dalam mendapatkan pengiraan yang tepat. Dalam kajian ini, proses pengiraan udang memerlukan masa yang panjang untuk mendapatkan jumlah bilangan amaun udang. Justeru, penggunaan komputer untuk mengira merupakan alternatif bagi pengiraan udang. Terdapat banyak pendekatan dalam proses pengiraan udang. Untuk menghasilkan satu rangka kerja yang tepat, pantas dan tidak memerlukan data yang banyak untuk proses pembelajaran, rangka kerja yang dicadangkan menggunakan Pengiraan Berdasarkan Pembelajaran Mendalam (*Deep Learning Based*) dengan algoritma Topeng R-CNN (*Mask R-CNN*). Sistem Pengiraan Udag ini dapat membawa kemudahan kepada penggunanya khususnya petani udang dan pembekal udang supaya dapat memendekkan masa dalam proses pengiraan udang.

KANDUNGAN

PENGAKUAN	II
PENGHARGAAN	III
ABSTRAK	IV
KANDUNGAN	V
SENARAI JADUAL	VIII
SENARAI ILUSTRASI	IX
SENARAI FORMULA	X
BAB I PENGENALAN	1
1.1 Pendahuluan	1
1.2 Pernyataan Masalah	1
1.3 Cadangan Penyelesaian	2
1.4 Objektif Kajian	2
1.5 Skop Kajian	3
1.6 Kekangan	3
1.7 Metod Kajian	3
1.8 Jadual	6
1.9 Kesimpulan	10
BAB II KAJIAN KESUSASTERAAN	11
2.1 Pengenalan	11
2.2 Latar Belakang Kajian	11
2.3 Kaedah Yang Digunakan Dalam Pengiraan Objek	11
2.3.1 Pengiraan Berdasarkan Pembelajaran Bukan Mesin	12
2.3.2 Pengiraan Berdasarkan Pembelajaran Mesin	12
2.3.3 Pengiraan Berdasarkan Pembelajaran Mendalam	12
2.3.3.1 Algoritma Topeng R-CNN (Mask R-CNN)	13
2.4 Kajian Lepas	15
2.4.1 Kaedah <i>Image Cutting, Binary(thresholding), Dilation Dan Enumeration</i>	15
2.4.2 Kaedah Rangkaian Neural Konvolusi	15
2.4.3 <i>Thresholding Otsu Dan Hough Transformation</i>	16
2.5 Cadangan Kajian	16
2.6 Perbandingan Kajian Sedia Ada Dan Yang Dicadang	18
2.7 Kesimpulan	18

BAB III	SPESIFIKASI KEPERLUAN	19
3.1	Pengenalan	19
3.2	Keperluan Pengguna Definisi	19
3.3	Ketentuan Sistem Spesifikasi	19
3.4	Model Sistem	20
3.5	Kesimpulan	21
BAB IV	SPESIFIKASI REKA BENTUK	22
4.1	Pengenalan	22
4.2	Proses Pembelajaran Aliran	22
4.3	Reka Bentuk Seni Bina	23
4.4	Penyediaan Set Data	24
4.5	Teknik Dan Algoritma	25
4.5.1	Data Pra-pemrosesan	26
4.5.2	Topeng R-CNN (<i>Mask R-CNN</i>)	26
4.5.3	Peraturan Ketepatan	27
4.6	Carta Aliran Bagi Proses Pengiraan Secara Manual	29
4.7	Carta Aliran Rangka Kerja	30
4.8	Reka Bentuk Pangkalan Data	31
4.9	Reka Bentuk Antara Muka	31
4.10	Kesimpulan	33
BAB V	PEMBANGUNAN SISTEM	34
5.1	Pengenalan	34
5.2	Pembangunan Sistem	34
5.2.1	Fasa Perolehan Perkakasan	35
5.2.2	Fasa Perolehan Set Data	36
5.2.3	Fasa Model Latihan	37
5.2.4	Fasa Implementasi	38
5.3	Kesimpulan	40
BAB VI	PERANCANGAN PENGUJIAN	41
6.1	Pengenalan	41
6.2	Reka Bentuk Untuk Prototaip	41
6.3	Proses Pengambilan Gambar	42
6.4	Data Ujian	43
6.5	Penetapan Parameter Analisis Untuk Model Latihan	44

6.6	Pengenalan Dengan <i>Hyperparameter</i>	46
6.7	<i>Hyperparameter</i> Untuk Diuji	46
6.7.1	<i>DETECTION_MAX_INSTANCE</i>	47
6.7.2	<i>MAX_GT_INSTANCE</i>	48
6.7.3	<i>RPN_NMS_THRESHOLD</i>	48
6.7.4	<i>RPN_TRAIN_ANCHORS_PER_IMAGE</i>	48
6.7.5	<i>TRAIN_ROIS_PER_IMAGE</i>	48
6.7.6	<i>VALIDATION_STEPS</i>	49
6.7.7	<i>STEPS_PER_EPOCH</i> Dan <i>NUMBERS OF EPOCH</i>	49
6.7.8	<i>REGULARIZATION</i>	50
6.7.9	<i>OPTIMIZERS</i>	51
6.7.10	<i>LEARNING RATE</i>	51
6.7.11	<i>BATCH SIZE</i>	51
6.7.12	<i>LEARNING MOMENTUM</i>	51
6.7.13	<i>WEIGHT DECAY</i>	52
6.8	Kesimpulan	52
BAB VII	HASIL KAJIAN	53
7.1	Pengenalan	53
7.2	Rangka Kerja Untuk Memilih <i>Hyperparameter</i> Yang Optimum	53
7.3	Parameter Analisis Yang Optimum Untuk Model Latihan	54
7.4	Hasil Keputusan Pengujian Dan Perbincangan	56
7.4.1	<i>Loss Function</i>	56
7.4.2	<i>Mean Average Precisions (mAP)</i>	58
7.4.3	<i>Precisions, Dan Recall</i>	58
7.4.4	Ketepatan Mengikuti Kategori	59
7.5	Peta Ketumpatan (<i>Density Map</i>)	61
7.6	Perbandingan Keputusan Antara Bilangan Sebenarnya Dengan Bilangan Tekaan	61
7.7	Kesimpulan	63
BAB VIII	KESIMPULAN	64
8.1	Pengenalan	64
8.2	Kekangan	64
8.3	Penambahbaikan Di Masa Hadapan	64
8.4	Kesimpulan	65
	RUJUKAN	66

Copyright@FTSM

SENARAI JADUAL

No. Jadual		Halaman
1.1	Grantt Chart bagi projek (Semester 1 2019/2020)	8
1.2	Grantt Chart bagi projek (Semester 2 2019/2020)	9
2.1	Perbandingan kajian sedia ada dan kajian yang dicadangkan	18
4.1	Nama atribut yang akan digunakan dalam sistem	31
6.1	Parameter analisis yang asal	46
7.1	Parameter analisis yang optimum untuk model latihan	56
7.2	Keputusan mAP bagi set data <i>train</i> dan <i>val</i>	58
7.3	Keputusan <i>Precisions</i> dan <i>Recall</i> bagi set data <i>train</i> dan <i>val</i>	59
7.4	Bilangan gambar berdasarkan jenis kepadatan dan bilangan undang	59
7.5	Keputusan peratusan ketepatan dan kesilapan berdasarkan jenis kepadatan	60
7.6	Keputusan purata ketepatan dan sisihan piawaian berdasarkan jenis kepadatan	60

SENARAI ILUSTRASI

No. Rajah		Halaman
1.1	Fasa-fasa dalam metodologi	4
1.2	Struktur pecahan kerja bagi projek pada Semester 1 2019/2020	6
1.3	Struktur pecahan kerja bagi projek pada Semester 2 2019/2020	7
2.1	Algoritma Topeng R-CNN (Mask R-CNN)	13
2.2	Algoritma Rangkaian Penyelesaian Sepenuhnya (Fully Convolutional Network-FCN)	14
3.1	Aliran model Topeng R-CNN (Mask R-CNN) untuk keseluruhan sistem	20
4.1	Proses pembelajaran aliran	22
4.2	Carta Hierarki Modul	23
4.3	Contoh gambar yang diambil dengan menggunakan kamera DSLR	24
4.4	Contoh gambar yang telah dilabelkan	25
4.5	Langkah umum pendekatan bagi fungsi Topeng R-CNN (Mask R-CNN)	26
4.6	Bekas yang digunakan dalam proses pengiraan secara manual	28
4.7	Carta aliran bagi proses pengiraan secara manual	29
4.8	Carta aliran rangka kerja	30
4.9	Antara muka untuk sistem	32
4.10	Antara muka selepas meletakkan model dan gambar	32
5.1	Gambar yang diambil menggunakan kamera DSLR	35
5.2	Gambar yang menunjukkan udang dalam keadaan kurang padat	36
5.3	Gambar yang menunjukkan udang dalam keadaan sederhana padat	36
5.4	Gambar yang menunjukkan udang dalam keadaan padat	37

5.5	Antara muka bagi Google Colab	37
5.6	Penghasilan model selepas fasa latihan model	38
5.7	Hasil dan keputusan bagi salah satu imej	39
5.8	Maklumat dan bilangan udang terdapat dalam imej	39
6.1	Reka bentuk untuk prototaip	41
6.2	Persediaan bagi proses pengambilan gambar	42
6.3	Pandangan dari atas bagi prototaip	42
6.4	Antara muka untuk proses pengambilan gambar	43
6.5	Sebahagian gambar yang simpan dalam folder val	44
6.6	Gambar yang mempunyai bilangan maksimum bagi bilangan tekaan	47
6.7	Gambar yang mempunyai bilangan maksimum bagi bilangan sebenar	47
6.8	Kotak yang dijana di salah satu imej bagi setiap tahap	48
6.9	Contoh ROIS yang dijanakan dalam gambar	49
7.1	Carta aliran untuk memilih hyperparameter yang optimum	53
7.2	Salah satu Loss untuk set data train	57
7.3	Salah satu Loss untuk set data val	58
7.4	Peta ketumpatan mengikut kepadatan udang	61
7.5	Regresi linear antara bilangan tekaan dan bilangan sebenar untuk 100 imej bagi model asal	62
7.6	Regresi linear antara bilangan tekaan dan bilangan sebenar untuk 100 imej bagi model yang selepas ditambahbaik	63

SENARAI FORMULA

No. Formula		Halaman
4.1	Formula untuk pengiraan peratusan ketepatan	28
6.1	Formula untuk Fungsi Kos (Cost Function)	45
6.2	Formula untuk Fungsi Kos (Cost Function) bagi L1	45
6.3	Formula untuk Fungsi Kos (Cost Function) bagi L2	46
7.1	Formula untuk banyak tugas (multi-task) Topeng R-CNN (Mask R-CNN)	54
7.2	Formula untuk jumlah keseluruhan bagi Loss	55
7.3	Formula untuk pengiraan bagi Precisions	57
7.4	Formula untuk pengiraan bagi Recall	57
7.5	Formula untuk pengiraan bagi peratusan ketepatan	58
7.6	Formula untuk pengiraan bagi peratusan kesilapan	58
7.7	Formula untuk pengiraan bagi sisihan piawaian	58
7.8	Formula untuk mengira nilai R^2	62

BAB I

PENGENALAN

1.1 PENDAHULUAN

Produk perikanan dapat menyumbang sejumlah besar permintaan protein kepada pemakanan manusia. Produk ikan juga merupakan salah satu makanan manusia di seluruh dunia. Merujuk kepada Laporan Petunjuk Pertanian Terpilih dalam Jabatan Statistik Malaysia, pengeluaran ikan bagi tahun 2015 adalah sebanyak 16.3% telah menurun sebanyak 5.6% kepada 10.7% pada tahun 2016. Manakala pengeluaran ikan bagi tahun 2017 adalah sebanyak 11.5% telah meningkat sebanyak 0.8% berbanding dengan tahun 2016. Walaupun peningkatan pengeluaran ikan, industri masih menghadapi kekurangan ikan berkualiti tinggi, ini telah mendorong syarikat ikan untuk ditubuhkan penetasan untuk mempercepat pengeluaran mereka. Pengeluaran ikan telah menjadi hambatan untuk pembangunan penternakan ikan di Malaysia dan proses pengiraan adalah salah satu masalah yang dihadapi oleh penetasan.

1.2 PENYATAAN MASALAH

Sebelum mewujudkan sistem pengiraan ikan dan udang, proses untuk mengira ikan dan udang menggunakan tenaga manusia untuk mengira tetapi terdapat proses pengiraan tersebut menggunakan masa yang panjang dan keputusan juga tidak tepat. Salah satu operasi yang penting dalam akuakultur adalah menghitung atau mengira. Ini sangat penting kerana ia membantu petani untuk merekod bilangan perikanan dengan tepat, menyediakan bilangan bekas yang tepat untuk menyimpan perikanan, menguruskan strategi makanan tepat dan mereka bentuk jadual pemasaran. Pengiraan bilangan ikan dan udang atau menyediakan anggaran yang tepat mengenai bilangan ikan dan udang adalah satu proses yang memakan masa bagi pelanggan. Ikan dan udang dikira dan disusun mengikut pelbagai saiz sebelum membekalkan ikan dan udang kepada petani berasaskan pada saiz mereka. Kaedah mengira dalam bidang perikanan di kawasan luar bandar kebanyakannya menggunakan kaedah mengira manual dengan tangan yang membawa tekanan kepada ikan dan udang serta kadang-kadang membawa kematian kepada ikan dan udang. Pemprosesan mengira manual adalah terdedah kepada kesilapan, peninggalan sekali-sekala serta keletihan.

1.3 CADANGAN PENYELESAIAN

Bagi menyelesaikan masalah tersebut, sebuah sistem yang membolehkan pengguna khususnya petani dan pembekal udang untuk menjalankan proses pengiraan dalam bidang perikanan akan dibangunkan. Pembangunan sistem ini akan dibangunkan dan bersifat seperti sistem pengiraan udang yang sudah ada tetapi akan lebih memesrakan pengguna. Sistem pengiraan udang ini akan mengandungi satu kategori, iaitu pengguna boleh menggunakan kaedah pemprosesan imej masa nyata iaitu memuat naik gambar. Sistem ini akan menggunakan kaedah pemprosesan imej untuk mengira bilangan udang yang terdapat di gambar.

1.4 OBJEKTIF KAJIAN

Objektif utama kajian ini adalah untuk mengira bilangan udang selepas memproses imej yang diperoleh daripada pangkalan data seperti gambar yang dimuat naik. Objektif kajian khususnya untuk mencapai objektif utama adalah:

- Membangunkan sistem pengiraan udang dengan menggunakan kaedah pemprosesan imej melalui gambar.
- Membangunkan sistem yang boleh mengira udang dengan menggunakan masa yang singkat.
- Membangunkan antara muka sistem pengiraan udang yang mesra pengguna terutamanya terdiri daripada golongan petani dan pembekal udang.

1.5 SKOP KAJIAN

Skop kajian ini fokus kepada golongan petani dan pembekal udang. Sistem yang akan dibangunkan ini berasaskan tidak memerlukan atas talian dan khas untuk bahasa pengaturcaraan Python sahaja. Sistem ini akan beroperasi menggunakan Windows. Sistem ini akan menggunakan dwibahasa supaya mudah difahami oleh semua pengguna dan mesra pengguna terutamanya golongan petani. Sempadan bagi produk akhir terbahagi kepada dua bahagian iaitu algoritma pemprosesan imej bagi objek udang akan dimasukkan dalam sistem ini dan objek selain daripada objek udang tidak akan dimasukkan dalam sistem. Ini menyebabkan sistem ini tidak dapat mengesan objek selain daripada objek udang.

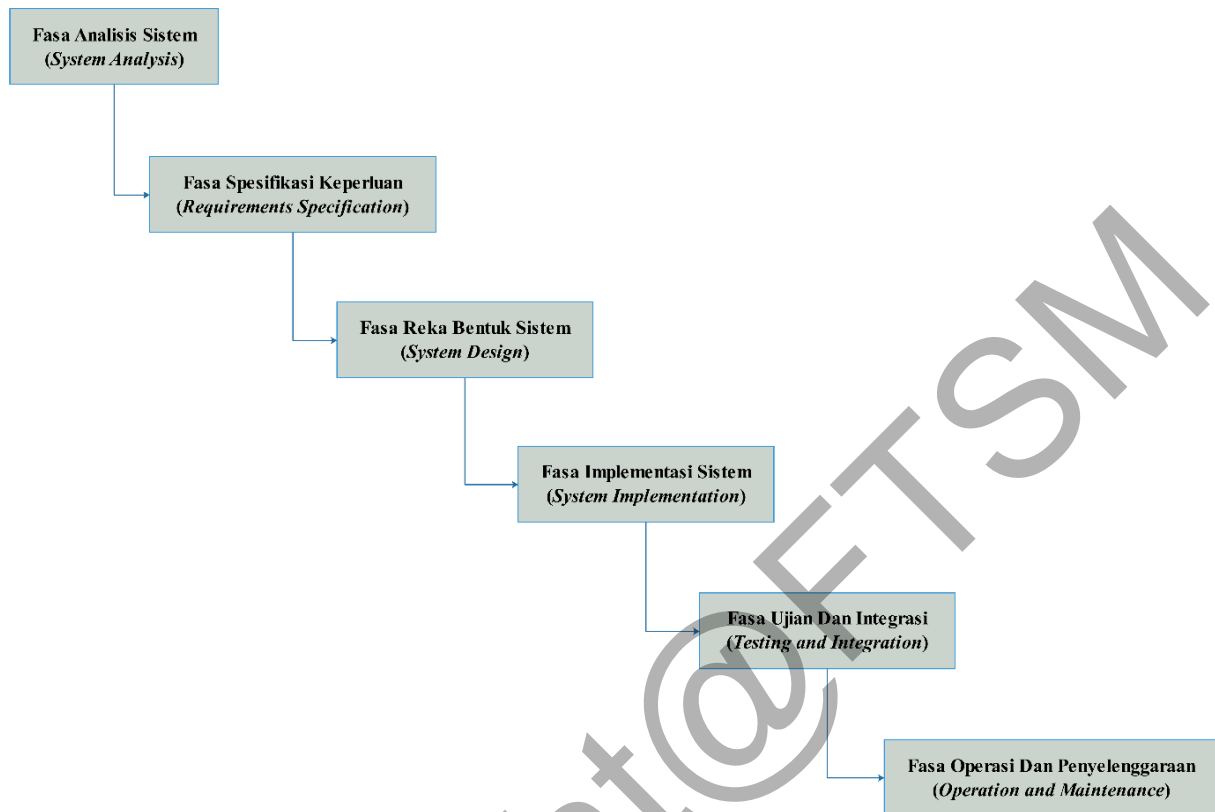
1.6 KEKANGAN

Kekangan yang dihadapi dalam sistem ini adalah seperti tidak dapat menyambung dengan pangkalan data (*database*) atas talian disebabkan saluran di tapak undang tersebut amat buruk. Selain itu, sistem ini mungkin tidak dapat menjalankan proses pengiraan dengan sebaik mungkin semasa terdapat undang yang saling berpotongan. Ini akan menyebabkan ketetapan pengiraan akan tidak tepat. Sistem ini juga hanya dapat menjalankan proses pengiraan sahaja, maklumat berkaitan suhu air, kelembapan air tidak dapat dibaca dengan menggunakan sistem ini.

1.7 METOD KAJIAN

Kaedah yang akan digunakan adalah Kitar Hayat Pembangunan Sistem (SDLC) di mana amalan *Waterfall* digunakan. Dalam amalan *Waterfall*, setiap fasa mesti siap sebelum fasa seterusnya bermula dan tidak ada pertindihan dalam fasa. Dalam amalan *Waterfall*, keseluruhan proses pembangunan perisian dibahagikan kepada fasa yang berasingan. Hasil daripada satu fasa bertindak sebagai input untuk fasa yang seterusnya secara berurutan. Ini bermakna mana-mana fasa dalam proses pembangunan bermula hanya fasa sebelumnya sudah selesai. Model *Waterfall* adalah proses reka bentuk berurutan di mana kemajuan dilihat sebagai mengalir ke bawah melalui fasa analisis sistem, fasa spesifikasi keperluan, fasa reka bentuk sistem, fasa implementasi sistem, fasa ujian dan integrasi dan fasa operasi dan penyelenggaraan.

Antara fasa-fasa yang dijalankan dalam kajian adalah:



Rajah 1.1 : Fasa-fasa dalam metodologi

I. Fasa Analisis Sistem (*System Analysis*)

Fasa ini melakukan pengamatan dan konsultasi dengan pengguna tentang perisian yang sudah dipakai selama ini dan keinginan yang diinginkan pengguna dalam proses pengembangan perisian ini.

II. Fasa Spesifikasi Keperluan (*Requirements Specification*)

Fasa ini melakukan konsultasi langsung dengan pengguna apa yang diinginkan secara spesifik dalam perisian ini. Fasa ini juga berkaitan dengan pemahaman apa yang perlu direka bentuk dan fungsi, tujuan dan lain-lain. Di sini, spesifikasi input dan output atau produk akhir dikaji dan ditandakan.

III. Fasa Reka Bentuk Sistem (*System Design*)

Fasa ini akan mengaji Fasa Spesifikasi Keperluan dari fasa kedua dan reka bentuk sistem akan disediakan. Reka bentuk sistem membantu dalam menentukan keperluan perkakasan dan juga membantu dalam mendefinisikan seni bina sistem keseluruhan. Kod perisian yang akan ditulis di peringkat seterusnya.

IV. Fasa Implementasi Sistem (*System Implementation*)

Fasa ini melakukan pembuatan dan pengerjaan terhadap perisian berdasarkan yang telah direka bentuk sebelumnya. Dengan input dari reka bentuk sistem, sistem ini akan dibangunkan dalam program yang kecil dipanggil unit, yang akan disatukan ke fasa seterusnya. Setiap unit dibangunkan dan diuji untuk fungsinya yang disebut sebagai *Unit Testing*.

V. Fasa Ujian Dan Integrasi (*Testing and Integration*)

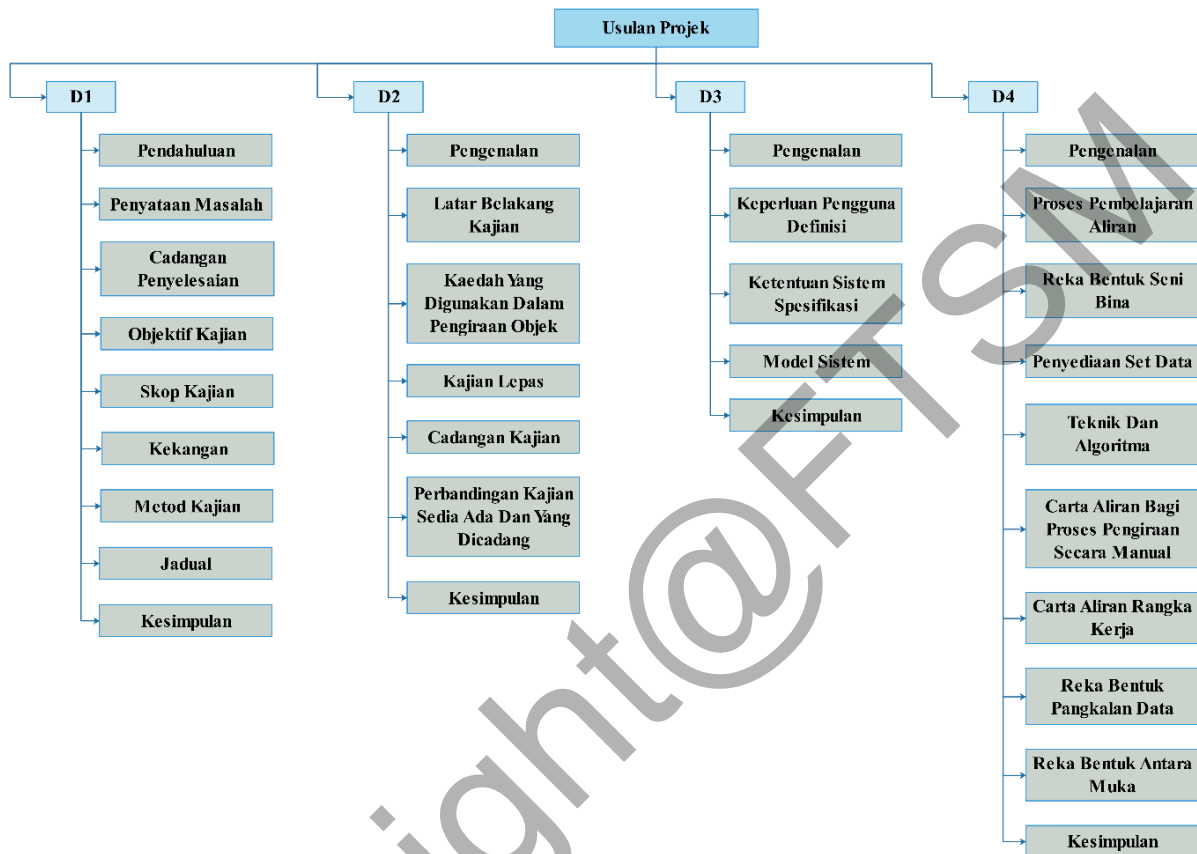
Semua unit yang dibangunkan dalam fasa implementasi sistem akan disatukan ke dalam sistem selepas ujian setiap unit. Perisian yang direka perlu melalui ujian perisian yang berterusan untuk mengetahui sama ada terdapat kecacatan atau kesilapan. Pengujian terhadap perisian atau sistem dilakukan supaya pengguna tidak menghadapi masalah semasa pemasangan perisian.

VI. Fasa Operasi Dan Penyelenggaraan (*Operation and Maintenance*)

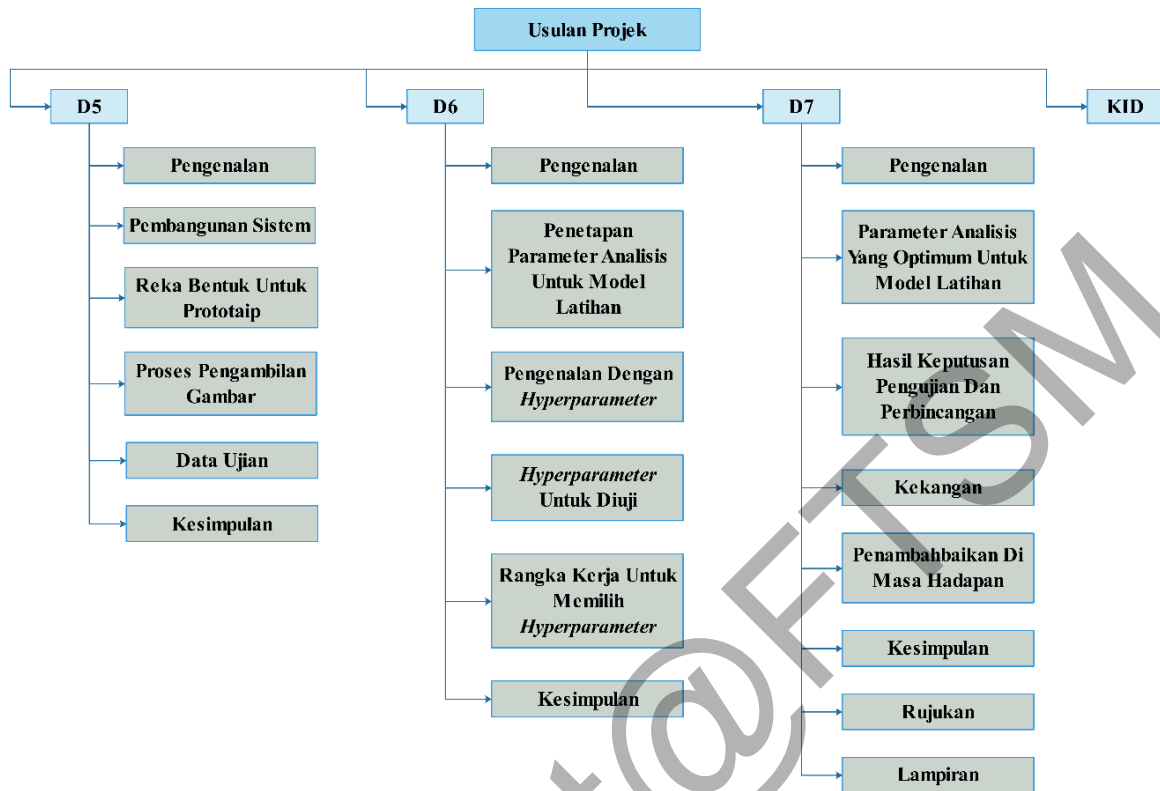
Fasa ini berlaku selepas pemasangan dan melibatkan membuat pengubahsuaian kepada sistem atau komponen individu untuk meningkatkan prestasi. Pengubahsuaian ini timbul sama ada disebabkan oleh perubahan permintaan yang dimulakan oleh pengguna, atau kecacatan yang ditemui semasa penggunaan secara langsung sistem. Pengguna disediakan dengan penyelenggaraan dan sokongan biasa untuk perisian yang dibangunkan.

1.8 JADUAL

Berikut merupakan jadual pelaksanaan yang dijangka untuk melaksanakan tugas sepanjang projek ini.



Rajah 1.2 : Struktur pecahan kerja bagi projek pada Semester 1 2019/2020



Rajah 1.3 : Struktur pecahan kerja bagi projek pada Semester 2 2019/2020

Minggu	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Modul 1 : Pengurusan Projek	■	■	■	■										
Perbincangan Pertama		■	■	■										
Modul 2 : Sorotan Susastera		■	■	■										
Perbincangan Ke-2			■	■										
Penghantaran D1				■										
Modul 3 : Kajian Keperluan				■	■	■								
Perbincangan Ke-3					■	■								
Penghantaran D2						■								
Modul 4 : Penulisan Ilmiah dan Teknikal							■	■	■	■				
Modul 5 : Kursus Reka Bentuk Perisian/Sistem							■	■	■	■				
Perbincangan Ke-4								■	■	■				
Perbincangan Ke-5									■	■				
Penghantaran D3										■				
Perbincangan Ke-6										■	■	■		
Kursus											■	■		
Penghantaran D4												■		
Perbincangan Ke-7													■	
Pra-KID														■

Jadual 1.1 : *Grantt Chart* bagi projek (Semester 1 2019/2020)

Minggu	1	2	3	4	5	6	7	8	9	10	11	12	13
Modul 1 : Pengurusan Projek	■	■											
Perbincangan Pertama	■	■											
Penghantaran D4 (Kemaskini)			■										
Demo I : Pemasangan Peralatan Pembangunan / Pemprosesan			■	■									
Demo II : Pembangunan / Pemprosesan					■								
Demo III : Pembangunan / Pemprosesan						■							
Penghantaran D5							■						
Model 2 : Pengujian Perisian, atau Penilaian Prestasi							■						
Perbincangan Ke-2								■					
Penghantaran D6									■				
Demo IV : Perlaksanaan Pengujian / Penilaian										■			
Perbincangan Ke-3											■		
Penghantaran D7												■	
KID													■

Jadual 1.2 : *Grantt Chart* bagi projek (Semester 2 2019/2020)

1.9 KESIMPULAN

Kesimpulannya, pemrosesan imej merupakan sesuatu bidang yang baru yang giat diterokai sekarang. Kajian-Kajian dijalankan adalah untuk memudahkan dan meningkatkan kecekapan kerja pengguna dalam pengiraan undang dan tidak memerlukan masa yang panjang untuk mengira bilangan undang.

Copyright@FTSM

BAB II

KAJIAN KESUSASTERAAN

2.1 PENGENALAN

Bab ini akan menerangkan kajian kesusasteraan yang berkaitan dengan kaedah yang digunakan untuk menjalankan proses pengiraan undang. Kajian-kajian kesusasteraan yang sedia ada adalah penting untuk dijadikan sebagai bahan rujukan kepada kajian ini. Kajian kesusasteraan dan kajian lepas yang berkaitan dengan pengiraan undang akan dibincangkan bagi membekalkan gambaran yang lebih teliti serta jelas terhadap topik ini. Selain itu, perbandingan antara kajian lepas turut akan dibincangkan.

2.2 LATAR BELAKANG KAJIAN

Sebagai contoh, misalnya suatu gambar yang dipenuhi dengan pemandangan yang kompleks atau merangkumi objek yang kompleks, masalah yang dihadapi adalah bagaimanakah untuk mengira jumlah objek seperti undang dalam media tersebut secara tepat. Merujuk kepada kaji selidik, banyak kemajuan penting yang telah dicapai dalam bidang pengiraan objek dalam imej (*Daniel Onoro-Rubio et al., 2016*). Pendekatan pembelajaran boleh dibahagikan kepada 2 kategori iaitu pembelajaran yang tanpa diselia (*Unsupervised Learning*) dan pembelajaran yang diselia (*Supervised Learning*). Salah satu kaedah yang dikategori dalam pembelajaran yang diselia adalah algoritma Topeng R-CNN (*Mask R-CNN*) dan algoritma ini akan dibincangkan dalam bahagian yang seterusnya.

2.3 KAEDAH YANG DIGUNAKAN DALAM PENGIRAAN OBJEK

Dalam proses pengiraan objek boleh dibahagikan kepada tiga kategori iaitu pengiraan berdasarkan pembelajaran bukan mesin, pengiraan berdasarkan pembelajaran mesin dan pengiraan berdasarkan pembelajaran mendalam.

2.3.1 Pengiraan Berdasarkan Pembelajaran Bukan Mesin (*Non-Machine Learning Based*)

Terdapat beberapa kaedah dalam pengiraan berdasarkan pembelajaran bukan mesin iaitu *Optical Flow*, pengiraan bilangan objek dengan warna. Berdasarkan kajian yang bertajuk “*Image recognition method using Local Binary Pattern and the Random forest classifier to count post larvae shrimp*” dijalankan oleh *Jirabhorn Kaewchote, Sittichoke Janyong dan Wasit Limprasert (Jirabhorn et al., 2018)*. Untuk menjalankan proses pengiraan larva, mereka menggunakan teknik *image recognition* dalam proses pengiraan tersebut. Terdapat dua kaedah *feature extraction* yang digunakan dalam proses pengiraan ini adalah *Local Binary Pattern (LBP)* dan *Red, Green, Blue (RGB) feature extraction* dengan *tree classifier, Random Forest (RF)*. Hasil daripada kajian ini menunjukkan bahawa teknik ini boleh mendapatkan ketepatan purata dengan sebanyak 98.5%.

2.3.2 Pengiraan Berdasarkan Pembelajaran Mesin (*Machine Learning Based*)

Pembelajaran mendalam mempunyai dua kategori iaitu pembelajaran yang tanpa diselia (*Unsupervised Learning*) dan pembelajaran yang diselia (*Supervised Learning*). Terdapat banyak algoritma dalam kategori pembelajaran yang diselia dan algoritma-algoritma tersebut boleh kepada dua kategori berdasarkan penggunaannya. Dua kategori penggunaan tersebut adalah penggunaan dalam klasifikasi (*classification*) dan penggunaan dalam regresi (*regression*). Berdasarkan kajian yang bertajuk “*Learning To Count Objects in Images*” dijalankan oleh *Victor Lempitsky dan Andrew Zisserman (Victor Lempitsky et al., 2010)*, kaedah yang digunakan dalam kajian tersebut adalah menggunakan pengiraan melalui Pengesanan (*Counting-By-Detection*), Pengiraan Melalui Regresi (*Counting-By-Regression*) dan Pengiraan Melalui Segmentasi (*Counting-By-Segmentation*).

2.3.3 Pengiraan Berdasarkan Pembelajaran Mendalam (*Deep Learning Based*)

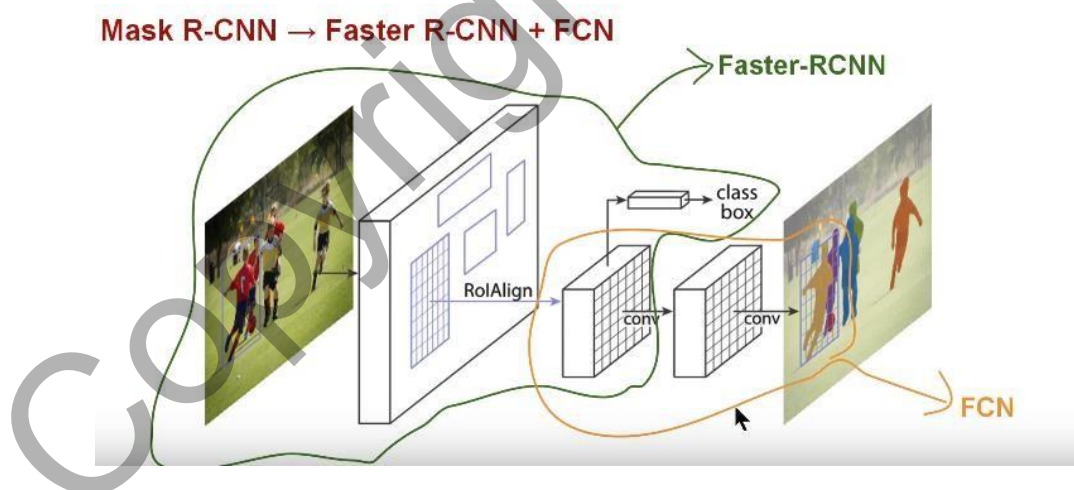
Pembelajaran mendalam adalah fungsi kecerdasan buatan yang meniru kerja otak manusia dalam memproses data dan mewujudkan corak untuk digunakan dalam membuat keputusan. Pembelajaran mendalam merupakan subset pembelajaran mesin dalam kecerdasan buatan yang mempunyai rangkaian yang mampu belajar tanpa pengawasan dari data yang tidak berstruktur atau tidak berlabel. Pembelajaran mendalam juga dikenali sebagai pembelajaran neural yang

mendalam (*Deep neural learning*) atau rangkaian neural yang mendalam (*Deep neural network*). Berdasarkan kajian yang bertajuk “*Divide and Count: Generic Object Counting by Image Divisions*” dijalankan oleh *Tobias Stahl, Silvia L. Pinte*a and *Jan C. van Gemert* (*Tobias Stahl et al., 2017*). Kaedah yang digunakan dalam kajian tersebut adalah menggunakan Rangkaian Neural Konvolusi (*Convolutional Neural Network-CNN*).

Dalam kajian ini akan menggunakan algoritma Topeng R-CNN (*Mask R-CNN*) untuk mendapatkan bilangan amaun udang. Topeng R-CNN (*Mask R-CNN*) juga merupakan salah satu kaedah pembelajaran mendalam. Bahagian seterusnya akan membincangkan tentang algoritma Topeng R-CNN (*Mask R-CNN*).

2.3.3.1 ALGORITMA TOPENG R-CNN (*Mask R-CNN*)

Segmentation Instance adalah lanjutan pengesanan objek, di mana topeng binari (iaitu objek vs. latar belakang) dikaitkan dengan setiap kotak yang terikat. Ini membolehkan maklumat yang lebih halus mengenai sejauh mana objek di dalam kotak. Terdapat beberapa algoritma yang melaksanakan segmentasi tetapi yang digunakan oleh *Tensorflow Object Detection API* adalah Topeng R-CNN (*Mask R-CNN*).

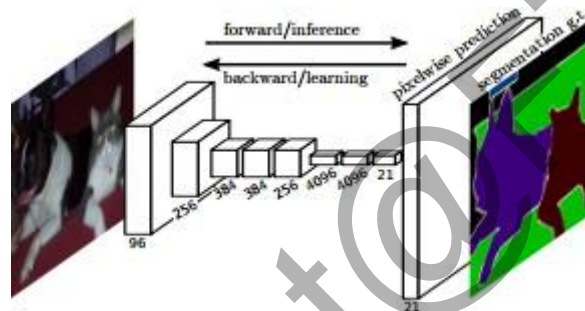


Rajah 2.1 : Algoritma Topeng R-CNN (*Mask R-CNN*). Adaptasi daripada (Kaiming He et al. 2018)

Faster R-CNN adalah algoritma yang sangat baik yang digunakan untuk pengesanan objek. *Faster R-CNN* terdiri daripada dua peringkat. Peringkat pertama, yang dipanggil Rangkaian Cadangan Wilayah (RPN), mencadangkan kotak calon objek yang dibatasi. Tahap

kedua, yang pada dasarnya *Fast R-CNN*, mengekstrak ciri-ciri menggunakan *RoIPool* dari setiap kotak calon dan melakukan pengklasifikasian dan regresi kotak-kotak. Ciri-ciri yang digunakan oleh kedua-dua peringkat boleh dikongsi untuk kesimpulan lebih cepat.

Topeng R-CNN (*Mask R-CNN*) secara konseptual mudah: *Faster R-CNN* mempunyai dua output bagi setiap objek calon, label kelas dan kotak pengikat yang terikat. Ini dapat menambah cawangan ketiga yang mengeluarkan topeng objek yang merupakan topeng binari yang menunjukkan piksel di mana objek berada di dalam kotak terikat. Tetapi output topeng tambahan berbeza dari output kelas dan kotak yang memerlukan pengekstrakan susunan spasial yang lebih halus bagi objek. Untuk melakukan Topeng R-CNN (*Mask R-CNN*) ini, akan menggunakan Rangkaian Penyelesaian Sepenuhnya (FCN) yang diterangkan di bawah.



Rajah 2.2 : Algoritma Rangkaian Penyelesaian Sepenuhnya (*Fully Convolutional Network-FCN*). Adaptasi daripada (Jonathan Long et al. 2015)

FCN adalah algoritma yang popular untuk melakukan segmentasi semantik. Model ini menggunakan pelbagai blok konvolusi dan lapisan kolam maksimum untuk menguraikan pertama imej ke $1/32$ dari saiz asalnya. Langkah yang kemudian akan membuat ramalan kelas pada tahap granulariti ini. Akhirnya akan menggunakan lapisan pensampelan dan dekonvolusi untuk mengubah saiz imej ke dimensi asalnya.

Pendek kata, Topeng R-CNN (*Mask R-CNN*) menggabungkan kedua-dua rangkaian iaitu terdiri daripada R-CNN dan FCN yang lebih cepat dalam satu seni bina mega. Fungsi kerugian untuk model adalah jumlah kerugian dalam membuat klasifikasi, menghasilkan kotak terikat dan menghasilkan topeng. Topeng R-CNN (*Mask R-CNN*) mempunyai beberapa penambahbaikan tambahan yang menjadikannya jauh lebih tepat daripada FCN.

2.4 KAJIAN LEPAS

Satu kajian lepas yang terdekat dengan kajian yang dicadangkan akan dibincangkan untuk memberi perbandingan dan gambaran yang lebih jelas tentang perbezaan kedua-dua kajian. Sistem yang sedia ada pada masa sekarang masih dalam tahap penyelidikan, ketepatan dalam proses pengiraan udang masih mempunyai peluang untuk menambahbaik dan mempertingkatkan.

2.4.1 Kaedah *Image Cutting, Binary (Thresholding), Dilation dan Enumeration*

Kajian yang bertajuk “*Vaname (Litopenaeus vannamei) Shrimp Fry Counting Based on Image Processing Method*” dijalankan oleh *M Solahudin, Slamet W dan Dwi A S (M Solahudin et al., 2018)*. Kaedah yang digunakan dalam kajian ini adalah *image cutting, binary (thresholding) and dilation* sebanyak tiga kali. Proses seterusnya adalah *enumeration*. *Enumeration* adalah penghitungan imej dengan menggunakan cara pelabelan dalam imej supaya dapat menghitung jumlah bilangan udang dalam imej tersebut. Keputusan program ujian dalam kajian ini menunjukkan ketepatan purata yang cukup tinggi iaitu 98.49%. Perbezaan antara kajian ini dengan kajian yang lain adalah penggunaan kamera digital dalam pengiraan. Penggunaan kamera digital dalam pelbagai bidang terbukti mampu menggantikan fungsi penglihatan manusia. Kelebihan teknologi ini dapat dilakukan kerja dengan cepat dan konsisten. Teknologi pemprosesan imej dijangka dapat membantu pengiraan udang dengan cepat, tepat dan murah. Purata masa yang digunakan dalam pengiraan amat pendek iaitu sebanyak 1.70 saat/imej berbanding dengan kajian yang lain memerlukan masa yang panjang.

2.4.2 Kaedah Rangkaian Neural Konvolusi (*Convolutional Neural Network-CNN*)

Berdasarkan kajian yang bertajuk “*Deep Count : Fruit Counting Based on Deep Simulated Learning*” dijalankan oleh *Maryam Rahneemofar dan Clay Sheppard (Maryam Rahneemofar et al., 2017)*. Kaedah yang digunakan dalam kajian ini adalah Rangkaian Neural Konvolusi (*Convolutional Neural Network-CNN*). Apabila mengetahui bilangan sebenar buah-buahan dan bunga dapat membantu petani membuat keputusan yang lebih baik mengenai amalan penanaman, penyakit tumbuhan pencegahan dan sebagainya. Pengiraan manual buah-buahan oleh pekerja adalah proses yang sangat memakan masa dan tidak praktikal. Algoritma ini dapat mengira dengan cekap walaupun berlaku pertindihan di antara buah-buahan.

Keputusan kajian ini menunjukkan 91% ketepatan ujian purata pada imej sebenar dan 93% pada imej sintetik. Perbezaan antara kajian ini dengan kajian yang lain adalah semua kaedah pengiraan berasaskan pembelajaran mendalam bergantung kepada pengesanan objek dan kemudian menghitung keadaan yang dikesan. Kaedah yang digunakan oleh *Maryam Rahneemoonfar* dan *Clay Sheppard* dapat menganggarkan kiraan objek dengan jelas dari pandangan seluruhnya gambar. Dengan cara ini, ia dapat mengira dengan jelas. Selain itu, teknik-teknik yang disebutkan di atas bergantung pada satu set besar data berlabel.

2.4.3 *Thresholding Otsu dan Hough Transformation*

Berdasarkan kajian yang bertajuk “*An Image Processing based Object Counting Approach for Machine Vision Application*” dijalankan oleh *Mehmet Baygin, Mehmet Karakose, Alisan Sarimaden* dan *Erhan Akin* (*Mehmet Baygin et al., 2018*). Kaedah yang digunakan dalam kajian ini adalah *thresholding Otsu* dan *Hough transformation*. Perbezaan antara kajian ini dengan kajian yang lain adalah menggunakan kaedah melakukan penghitungan automatik bebas daripada warna. Kajian ini juga memerlukan kamera untuk menghantar imej yang diambil dan algoritma pemprosesan imej akan digunakan untuk memproses imej. Hasil daripada kajian yang dilakukan adalah penyelesaian tersebut memberikan keputusan yang tepat dan masa yang singkat untuk mendapat keputusan.

2.5 CADANGAN KAJIAN

Berdasarkan objektif kajian yang ditetapkan, sebuah rangka kerja akan dihasilkan untuk mengira undang dalam imej secara tepat, pantas dan hanya memerlukan data yang selidik. Justeru, kaedah pembelajaran ketumpatan digunakan dan untuk mempercepatkan kaedah ini, algoritma Topeng R-CNN (*Mask R-CNN*) akan digunakan dalam kajian ini untuk menjalankan proses pengiraan undang. Ringkasan tentang algoritma yang digunakan akan diterangkan di bahagian seterusnya.

Kaedah pembelajaran ketumpatan juga akan digunakan dalam kajian ini yang dikaji oleh *Victor Lempitsky* dan *Andrew Zisserman*. Kaedah pembelajaran ketumpatan merupakan salah satu kaedah dalam kategori pembelajaran yang diselia dan serupa dengan kaedah pengiraan melalui regresi. Objektif kaedah ini adalah untuk mendapatkan pemetaan ciri-ciri

imej latihan supaya pemetaan tersebut boleh digunakan ke atas imej baru. Pemetaan ciri-ciri ke atas imej akan memberikan jumlah objek pada imej.

Sebelum memulakan proses pembelajaran, piksel iaitu elemen paling asas bagi imej perlu didapatkan. Setiap sel dalam imej latihan perlulah ditandakan supaya lokasi sel boleh dikenal pasti. Pembelajaran kemudian boleh dilaksanakan melalui pemetaan ciri-ciri dengan menggunakan lokasi sel pada imej dan jumlah objek pada imej. Selepas mendapatkan pemetaan ciri-ciri melalui pembelajaran, maka jumlah objek bagi imej baru boleh diperolehi dengan pemetaan ciri ke atas setiap ciri-ciri vektor imej baru. Kaedah ini mampu memberikan pengiraan objek yang tepat dan hanya memerlukan data yang sedikit.

Kajian ini juga akan menggunakan algoritma Topeng R-CNN (*Mask R-CNN*). Terdapat empat langkah dalam algoritma R-CNN. Langkah yang pertama, imej akan input ke dalam sistem seterusnya akan menggunakan algoritma *Selective Search* untuk mengesan lokasi imej yang berpotensi mengandungi objek. Kemudian, gunakan pembelajaran pemindahan, khususnya ciri pengekstrakan, untuk mengira ciri-ciri untuk setiap cadangan dengan menggunakan CNN yang telah terlatih. Langkah yang terakhir adalah klasifikasi setiap proposal menggunakan ciri-ciri yang diekstrak dengan *Support Vector Machine* (SVM).

Tetapi, algoritma R-CNN menggunakan masa yang panjang untuk mendapatkan keputusan. Untuk menambahbaik algoritma R-CNN, *Girshick et al.*, telah mencipta algoritma *Fast R-CNN*. Dua sumbangan utama telah membina dalam algoritma *Fast R-CNN* dan seterusnya dinamakan sebagai Topeng R-CNN (*Mask R-CNN*). Dua sumbangan utama adalah menggantikan modul *Pooling ROI* dengan modul *Align ROI* yang lebih tepat dan memasukkan cawangan tambahan daripada modul *Align ROI*.

Dalam set data yang diberi, akan mengesan objek seperti udang, dan seterusnya setiap set data akan terbahagi kepada beberapa bahagian. Dalam setiap bahagian akan mengesan dan bulatkan lokasi yang ingin dilatih oleh model supaya ketepatan yang didapati akan lebih tinggi.

2.6 PERBANDINGAN KAJIAN SEDIA ADA DAN YANG DICADANG

	<i>Deep Count : Fruit Counting Based on Deep Simulated Learning (Maryam Rahnemoonfar et al. 2017)</i>	Rangka Kerja Yang Dicapang
1. Perisian	-	<i>Google Colab</i>
2. Kaedah	Pembelajaran Mendalam	Pembelajaran Mendalam
3. Bahasa Pengaturcaraan	<i>Python</i>	<i>Python</i>
4. Algoritma	Rangkaian Neural Konvolusi (<i>Convolutional Neural Network-CNN</i>)	Topeng R-CNN (<i>Mask R-CNN</i>)

Jadual 2.1 : Perbandingan kajian sedia ada dan kajian yang dicadangkan

Berdasarkan jadual di atas, perisian yang digunakan dalam kajian lepas yang dikaji tidak diketahui dan kajian yang dicadangkan akan menggunakan *Google Colab* sebagai perisian. Kajian lepas dan kajian yang dicadangkan juga menggunakan kaedah pembelajaran mendalam sebagai kaedah. Kedua-dua kajian menggunakan bahasa pengaturcaraan Python. Bagi algoritma dalam kajian lepas menggunakan Rangkaian Neural Konvolusi (*Convolutional Neural Network-CNN*) manakala kajian yang dicadangkan menggunakan algoritma Topeng R-CNN (*Mask R-CNN*).

2.7 KESIMPULAN

Melalui kajian literasi, kaedah yang sedia ada boleh difahami dengan teliti. Kajian-kajian lepas ini juga telah memberikan banyak pengetahuan dan pandangan daripada ramai pengkaji. Justeru, perbandingan dan perbezaan antara kajian akan dijadikan sebagai rujukan semasa proses pembangunan kajian ini. Hasil dan kod kajian lepas juga akan diambil sebagai rujukan dalam pengekodan dan pengujian. Algoritma Topeng R-CNN (*Mask R-CNN*) juga menunjukkan kecekapan berbanding dengan algoritma yang lain kerana algoritma ini menggunakan masa yang lebih pendek berbanding dengan algoritma lain. Algoritma ini dipilih untuk diubahsuai dalam projek ini kerana mempunyai ruang untuk penambahbaikan.

BAB III

SPESIFIKASI KEPERLUAN

3.1 PENGENALAN

Bab ini akan menerangkan spesifikasi keperluan. Spesifikasi keperluan terbahagi kepada dua bahagian iaitu keperluan pengguna dan keperluan sistem. Spesifikasi keperluan berkait rapat dengan keperluan sistem dan pengguna. Spesifikasi keperluan memainkan peranan penting dalam proses pembangunan sistem supaya kajian dapat dijalankan lebih lancar. Spesifikasi keperluan perlu dicapai sebelum menggunakan algoritma bagi memastikan proses dapat dijalankan tanpa ralat sistem.

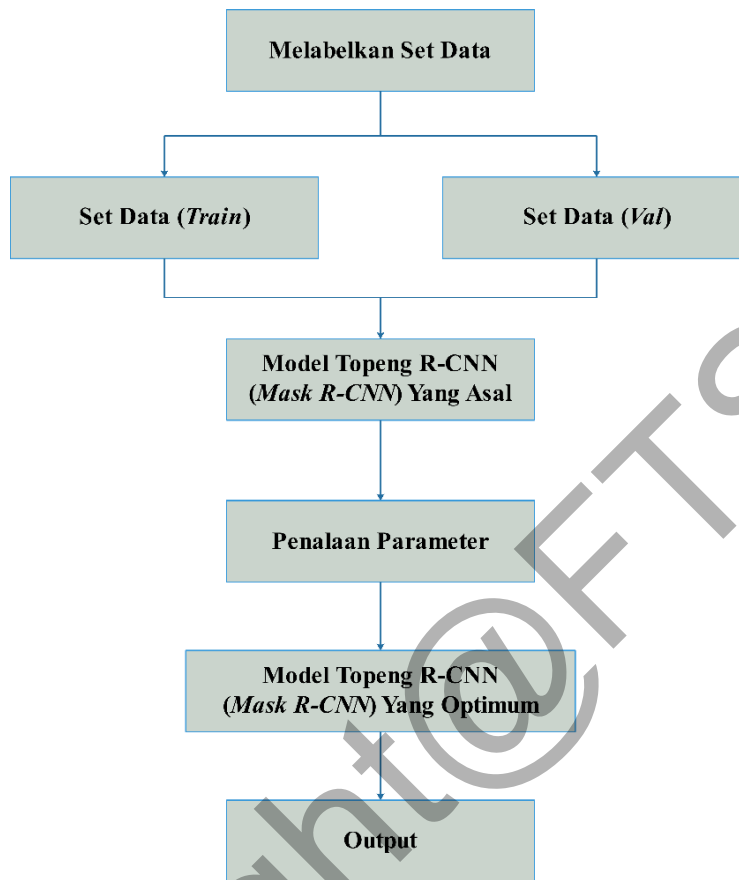
3.2 KEPERLUAN PENGGUNA DEFINISI

- Sistem ini harus mengesan objek seperti udang dalam gambar yang dimuat naik.
- Sistem ini harus mendapatkan bilangan udang yang akan dikira.
- Sistem ini harus membulatkan objek udang yang terdapat dalam gambar yang dimuat naik.

3.3 KETENTUAN SISTEM SPESIFIKASI

- Pengguna patut diberikan kemudahan untuk membolehkan memuat naik gambar yang merangkumi udang ke dalam sistem.
- Pengguna patut diberikan kemudahan untuk mendapatkan bilangan anggaran udang yang terdapat dalam gambar yang dimuat naik.
- Pengguna patut diberikan kemudahan untuk mendapatkan output seperti gambar yang sudah membulatkan objek udang.

3.4 MODEL SISTEM



Rajah 3.1 : Aliran model Topeng R-CNN (*Mask R-CNN*) untuk keseluruhan sistem

Rajah 3.1 menunjukkan aliran model Topeng R-CNN (*Mask R-CNN*) untuk keseluruhan sistem. Topeng R-CNN (*Mask R-CNN*) adalah model segmentasi contoh yang dapat mengenal pasti piksel dengan lokasi piksel tidak kira objek tersebut di mana-mana kedudukan. Model ini menerima set latihan juga dikenali sebagai set data (*train*) untuk pembelajaran set data. Kemudian kecekapan dan ketepatan diuji dengan set data (*val*).

Latihan model akan terlibat dalam proses melatih objek pengesanan model disebabkan topeng objek juga diperlukan semasa proses melatih set data dijalankan. Penalaan parameter akan dilakukan sekiranya keputusan kurang memuaskan. Akhir sekali, set parameter yang memberi ketepatan dan kecekapan yang paling bagus akan digunakan untuk menguji set data yang lain.

3.5 KESIMPULAN

Kesimpulannya, ketentuan spesifikasi bagi pengguna dan sistem amat penting dalam proses menciptakan sesuatu sistem. Seperti dalam kajian ini, matlamat yang perlu dicapai oleh pengguna adalah dapat menjalankan proses pengiraan dengan menggunakan sistem ini untuk mendapatkan bilangan anggaran undang.

Copyright@FTSM

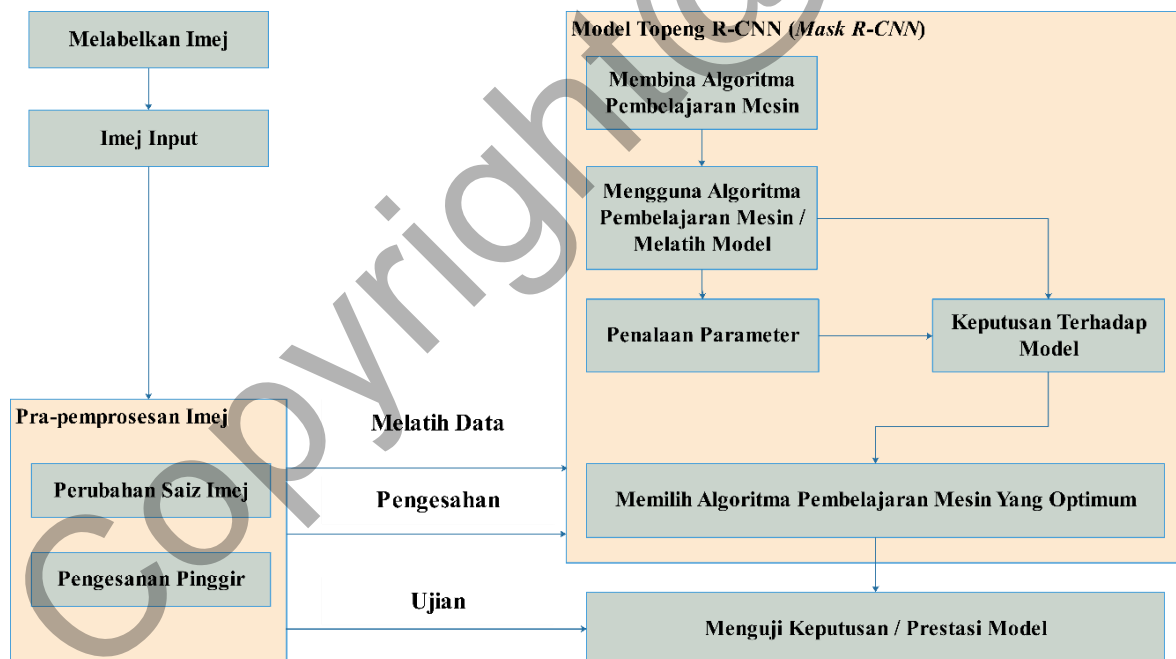
BAB IV

SPESIFIKASI REKA BENTUK

4.1 PENGENALAN

Metodologi dan reka bentuk akan dibincangkan dalam bahagian ini. Bab ini akan diterangkan secara terperinci melalui rangka kerja dan algoritma yang dicadangkan. Dengan rangka kerja yang jelas, proses kajian dan pengkodan dapat dijalankan dengan lebih lancar. Bab ini penting kerana ia merupakan asas bagi perancangan dalam fasa pelaksanaannya yang akan dijalankan kemudian.

4.2 PROSES PEMBELAJARAN ALIRAN

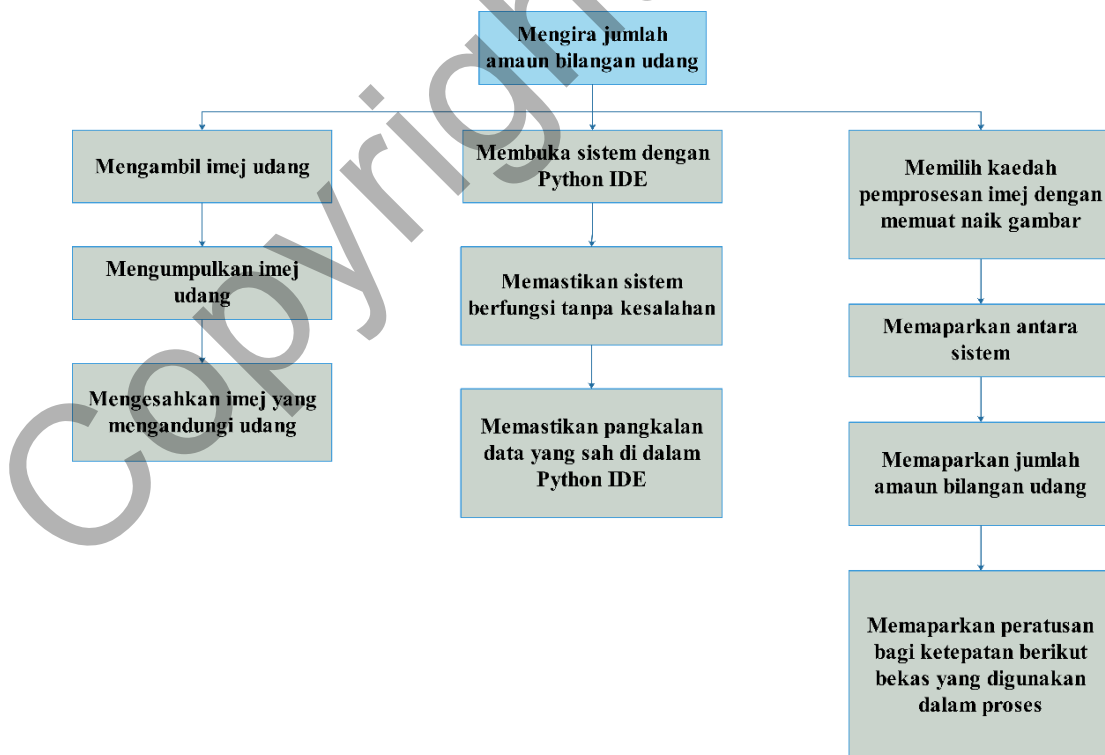


Rajah 4.1 : Proses pembelajaran aliran

Rajah 4.1 di atas menunjukkan bahawa projek tahun akhir ini akan mengalami proses melabelkan set data yang telah dijelaskan pada Bab 3 Spesifikasi Keperluan dalam bahagian model sistem (Rujuk Rajah 3.1). Selepas itu, ciri yang diekstrak akan dibahagikan kepada 3 proses iaitu latihan data, pengesahan dan proses ujian. Pertama, data latihan dengan ciri yang diekstrak akan menggunakan algoritma pembelajaran mesin dalam kajian ini. Kedua, proses pengesahan digunakan untuk menukar penalaan parameter yang sesuai ke dalam algoritma. Ini adalah bahagian cuba dan ralat untuk mencari hasil yang paling tepat. Bahagian terakhir daripada tiga proses itu adalah proses pengujian. Algoritma dengan parameter yang optimum akan dipilih dan digunakan untuk mencuba imej lain untuk menguji hasilnya serta prestasi model tersebut.

4.3 REKA BENTUK SENI BINA

Reka Bentuk Seni Bina yang digunakan dalam projek tahun akhir ini adalah Carta Hierarki Modul. Carta Hierarki Modul digunakan untuk mengilustrasikan modul atau sub-sistem dan hubungan mereka. Rajah 4.2 menunjukkan Carta Hierarki Modul projek tahun akhir ini.



Rajah 4.2 : Carta Hierarki Modul

4.4 PENYEDIAAN SET DATA

Dalam projek tahun akhir ini, data telah diambil dengan menggunakan kamera DSLR bermodel Canon EOS 80D. Mod kamera yang digunakan adalah C1. Apertur (*Aperture*) adalah f/5.6, kelajuan pengatup (*Shutter Speed*) adalah 1/100 saat, ISO 100, kanta kamera (*Lens Camera*) yang digunakan adalah 18-55 mm dan jarak untuk mengambil gambar adalah 10 cm.



Rajah 4.3 : Contoh gambar yang diambil dengan menggunakan kamera DSLR

Gambar yang diambil akan mengubah saiz imej daripada 6000 piksel bagi lebar dan 4000 piksel bagi ketinggian kepada 960 piksel bagi lebar dan 640 piksel bagi ketinggian supaya dapat memudahkan kerja semasa melabelkan udang dalam gambar. Kemudian, udang yang dalam gambar ini akan dilabelkan dengan menggunakan *VGG Image Annotator* (Rujuk Rajah 4.4). Imej input berwarna akan melalui pra-pemprosesan dalam algoritma mengubah saiz imej semula dan pengesanan pinggir sebelum menjalankan fasa melatih data, fasa pengesanan dan fasa ujian.

Imej input mestilah mempunyai ciri-ciri yang dinyatakan di bawah untuk memastikan ketepatan dalam proses pengiraan.

Ciri 1 : Imej yang digunakan adalah imej yang mengandungi udang.

Ciri 2 : Minimum saiz imej adalah 800 piksel manakala maximum saiz adalah 1024 piksel termasuk lebar dan ketinggian.

Ciri 3 : Resolusi imej mestilah jelas.



Rajah 4.4 : Contoh gambar yang telah dilabelkan

4.5 TEKNIK DAN ALGORITMA

Projek tahun akhir ini akan bertumpu kepada teknik dan algoritma seperti data pra-pemprosesan, Topeng R-CNN (*Mask R-CNN*) dan pelbagai cara untuk menguji prestasi model termasuk peratusan ketepatan.

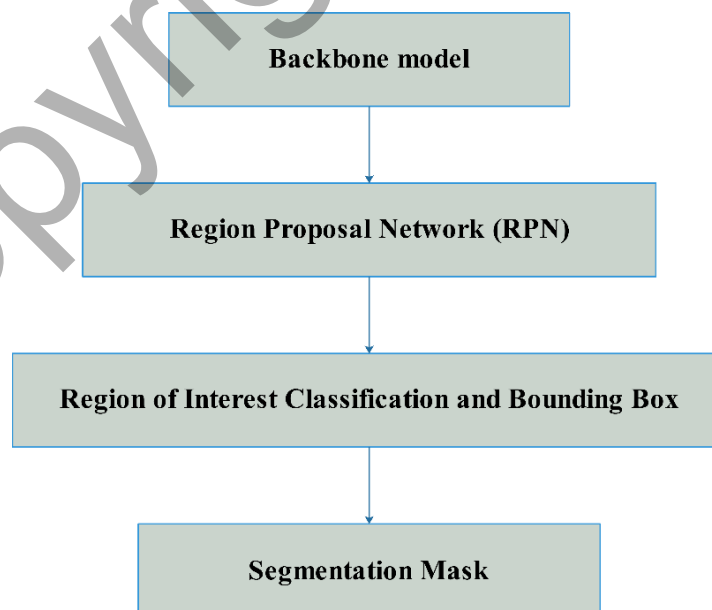
4.5.1 Data Pra-pemrosesan

Pra-pemrosesan Imej seperti menukarkan perubahan saiz imej dan pengesanan pinggir merupakan proses yang amat penting untuk mendapatkan keputusan yang lebih tepat. Selain itu, pra-pemrosesan ini juga penting bagi fasa melatih data dan fasa pengesanan yang akan dijalankan selepas ini.

4.5.2 Topeng R-CNN (*Mask R-CNN*)

Topeng R-CNN (*Mask R-CNN*) adalah lanjutan daripada *Faster R-CNN*. *Faster R-CNN* meramalkan kotak terikat dan Topeng R-CNN (*Mask R-CNN*) pada dasarnya menambah satu lagi cawangan untuk meramalkan topeng objek secara selari.

Topeng R-CNN (*Mask R-CNN*) adalah teknik segmentasi contoh yang menempatkan setiap piksel yang terdapat di setiap objek dalam imej dan bukannya kotak sempadan. Ia mempunyai dua peringkat : cadangan rantau dan kemudian mengklasifikasikan cadangan dan menghasilkan kotak dan topeng. Ia menggunakan rangkaian tambahan konvolusi sepenuhnya di atas peta ciri berasaskan CNN dengan input sebagai peta ciri dan memberi matriks dengan 1 pada semua lokasi di mana piksel dimiliki objek dan 0 di tempat lain sebagai output.



Rajah 4.5 : Langkah umum pendekatan bagi fungsi Topeng R-CNN (*Mask R-CNN*)

Berdasarkan Rajah 4.5, berikut adalah langkah-langkah umum pendekatan bagi fungsi Topeng R-CNN (*Mask R-CNN*).

Langkah 1 : *Backbone model*

Rangkaian neural convolutional standard yang berfungsi sebagai pengekstraksi ciri. Sebagai contoh, ia akan menjadikan imej 1024x1024x3 menjadi peta ciri 32x32x2048 yang berfungsi sebagai input untuk lapisan seterusnya.

Langkah 2 : Rangkaian Cadangan Wilayah (*Region Proposal Network*) (RPN)

Menggunakan rantau yang ditakrifkan dengan sebanyak 200K kotak penahan, RPN mengimbas setiap rantau dan meramalkan sama ada objek itu ada atau tidak. Salah satu kelebihan utama RPN adalah ia tidak mengimbas imej sebenar, rangkaian akan mengimbas peta ciri, menjadikannya lebih cepat.

Langkah 3 : *Region of Interest Classification and Bounding Box*

Dalam langkah ini, algoritma mengambil rantau yang dicadangkan oleh RPN sebagai input dan mengeluarkan klasifikasi (*softmax*) dan sebuah kotak (*regressor*).

Langkah 4 : *Segmentation Masks*

Pada peringkat akhir, algoritma rantau ROI positif akan diambil sebagai input dan topeng 28x28 piksel dengan nilai apungan akan dijana sebagai output untuk objek. Semasa kesimpulan, skala topeng ini akan ditingkatkan.

4.5.3 Peratusan ketepatan

Bagi pengiraan peratusan ketepatan, berikut adalah formula (Rujuk Formula 4.1) untuk mengadakan pengiraan peratusan ketepatan.

$$\frac{\begin{array}{l} \square\square\square\square\square\square \\ \square\square = \end{array}}{\begin{array}{l} \square\square\square\square\square\square \\ \square\square\square\square\square \\ \hline \square\square\square\square\square\square \\ \square\square\square\square\square\square \end{array}} \times 100\%$$

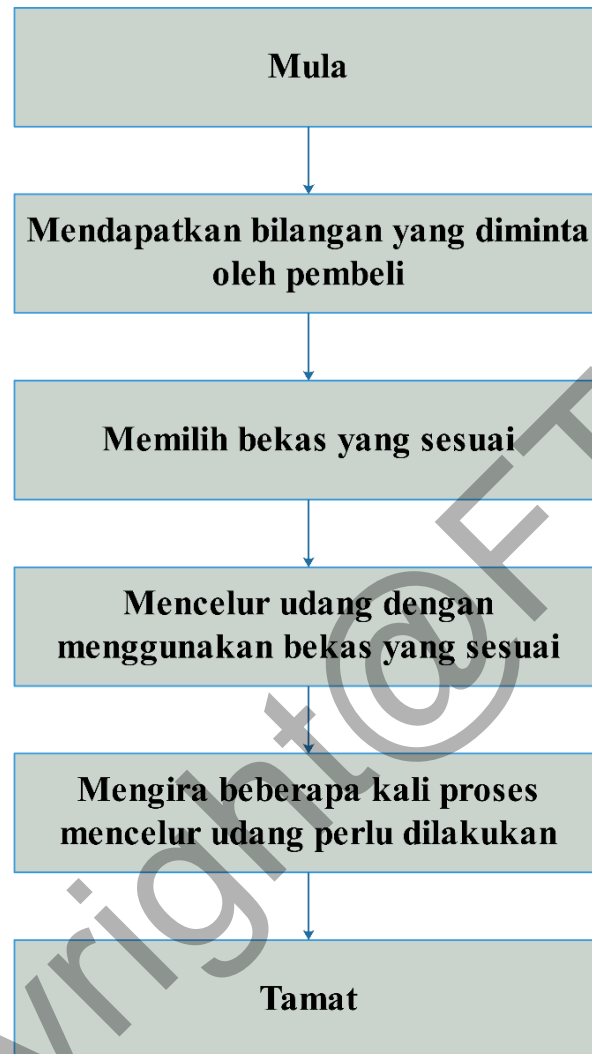
Formula 4.1 : Formula untuk pengiraan peratusan ketepatan



Rajah 4.6 : Bebas yang digunakan dalam proses pengiraan secara manual

Dalam rajah 4.6, setiap bebas mempunyai jumlah amaun bilangan undang yang telah ditetapkan. Untuk mendapat peratusan ketepatan, jumlah amaun bilangan undang yang dapat di sistem akan dibandingkan dengan jumlah amaun bilangan undang yang ditetapkan oleh bebas tersebut.

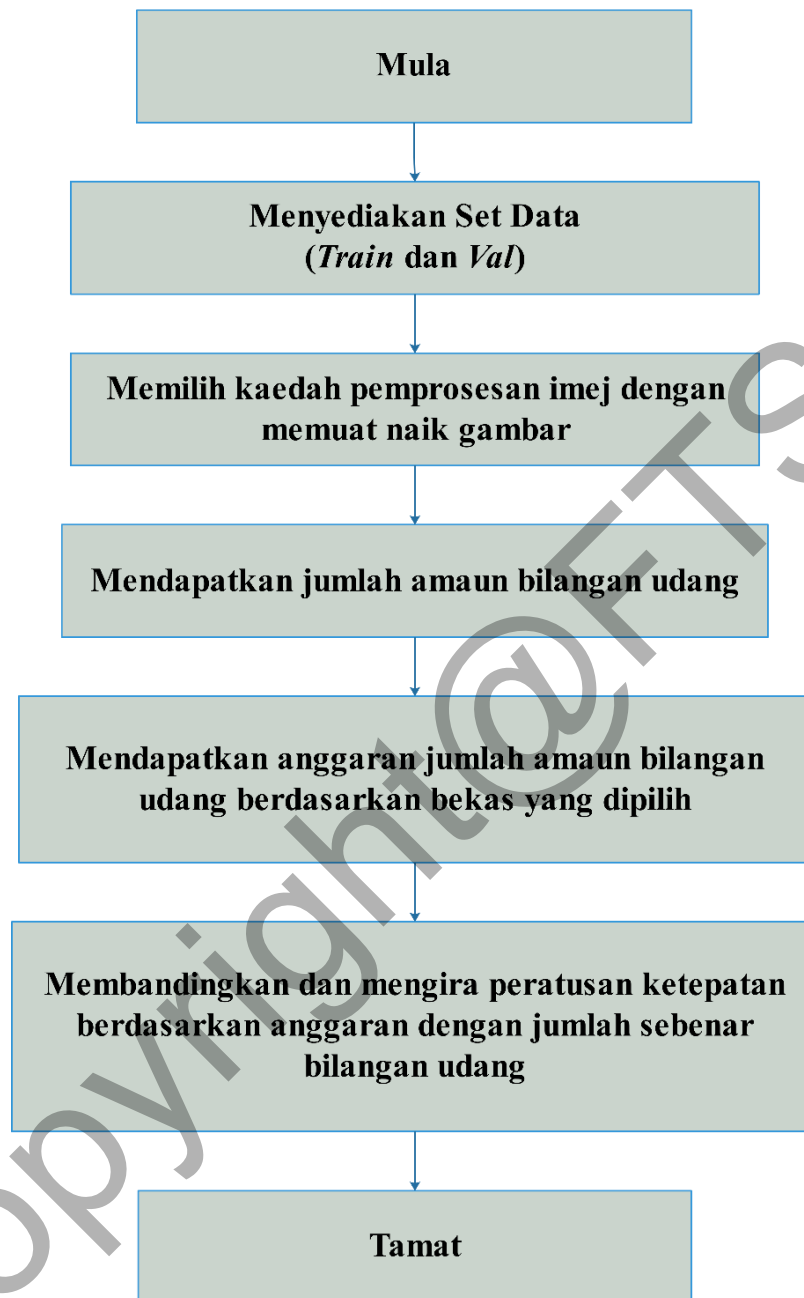
4.6 CARTA ALIRAN BAGI PROSES PENGIRAAN SECARA MANUAL



Rajah 4.7 : Carta aliran bagi proses pengiraan secara manual

Berdasarkan rajah 4.7, langkah pertama adalah pembekal perlu mendapatkan bilangan yang diminta oleh pembeli. Langkah yang seterusnya, pembekal akan memilih bekas yang disesuaikan berdasarkan bilangan amaun yang diminta oleh pembeli. Pembekal akan memulakan proses mencelur udang dengan menggunakan bekas yang sesuai dan letak udang tersebut di dalam plastik beg. Langkah akhir adalah pembekal akan mengira beberapa kali proses mencelur udang perlu dilakukan sehingga jumlah bilangan amaun udang sudah mencukupi.

4.7 CARTA ALIRAN RANGKA KERJA



Rajah 4.8 : Carta aliran rangka kerja

Berdasarkan rajah 4.8, carta aliran rangka kerja disediakan bagi memberikan gambaran tentang rangka kerja dengan lebih jelas dan mudah. Pada peringkat permulaan, set data iaitu imej akan disediakan. Pengguna boleh memilih kaedah pemprosesan imej dengan memuat naik gambar. Pengguna akan mendapatkan jumlah amaun bilangan udang yang dikeluarkan oleh sistem.

Pengguna perlu mendapatkan anggaran jumlah amaun bilangan udang berdasarkan bekas yang dipilih supaya dapat membandingkan dan mengira peratusan ketepatan berdasarkan anggaran dengan jumlah sebenar bilangan udang.

4.8 REKA BENTUK PANGKALAN DATA

Antara nama atribut yang akan digunakan dalam sistem tersebut adalah seperti berikut :

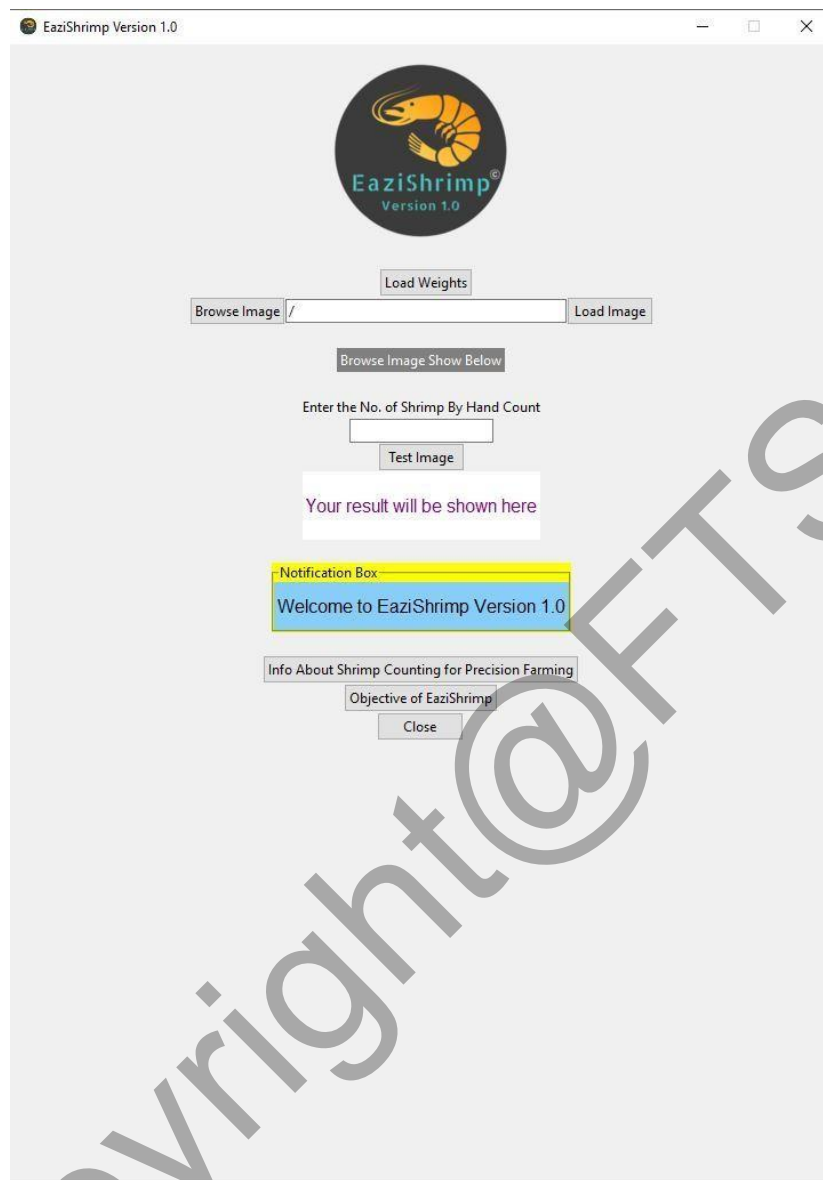
Nama Atribut	Jenis	Keterangan
Amount Set By Container / Jumlah Yang Ditetapkan Oleh Bekas	<i>Integer</i>	Jumlah bilangan udang yang ditetapkan oleh bekas
Total / Jumlah	<i>Integer</i>	Jumlah bilangan udang yang dikira oleh sistem
Accuracy / Ketepatan	<i>Double</i>	Peratusan bagi ketepatan berikut bekas yang digunakan dalam proses pengiraan udang secara manual berbanding dengan jumlah amaun bilangan udang yang dikira oleh sistem

Jadual 4.1 : Nama atribut yang akan digunakan dalam sistem

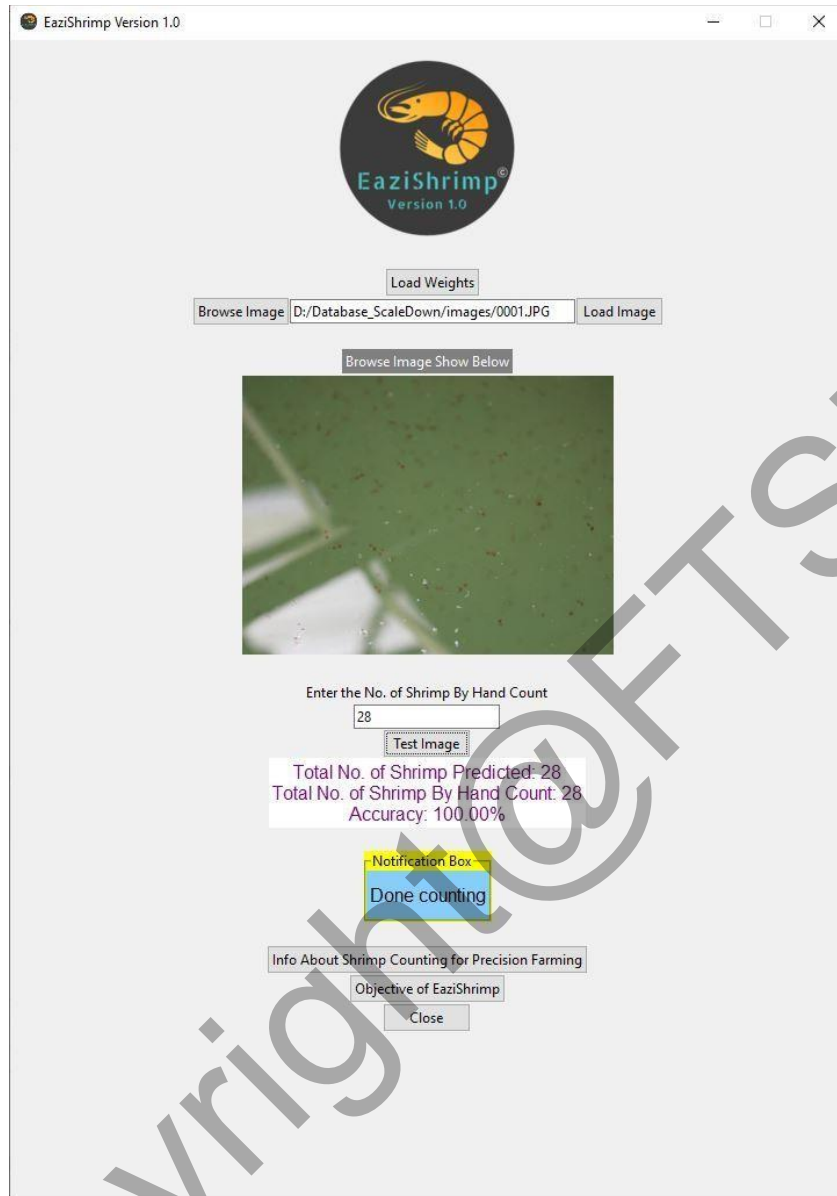
4.9 REKA BENTUK ANTARA MUKA

Perisian Visual Studio Code akan digunakan untuk membangunkan antara muka pengguna grafik atau *Graphical User Interface* (GUI) bagi projek sistem pengiraan udang ini. Rajah-rajah di bawah merupakan antara muka pengguna grafik akan dibangunkan bagi projek tahun akhir ini.

Dalam antara muka mempunyai 6 bahagian, iaitu bahagian untuk meletakkan model, bahagian untuk meletakkan gambar, bahagian untuk memaparkan gambar, bahagian untuk bermula pengiraan, bahagian untuk memaparkan bilangan udang dan kotak pemberitahuan.



Rajah 4.9 : Antara muka untuk sistem



Rajah 4.10 : Antara muka selepas meletakkan model dan gambar

4.10 KESIMPULAN

Spesifikasi reka bentuk dan rangka kerja yang dicadangkan diterangkan dalam bab ini. Spesifikasi reka bentuk yang dibincangkan dapat bermanfaat kepada proses reka bentuk dan fungsi sistem pengiraan udang. Gambaran yang lebih jelas boleh dipelajari melalui bab ini. Pada masa yang sama, meningkatkan lagi kefahaman terhadap rangka kerja yang dicadangkan dan mengenal pasti sama ada masalah yang mungkin timbul daripadanya. Dengan merujuk kepada spesifikasi reka bentuk, hasil reka bentuk dapat memenuhi objektif yang dibincangkan dalam Bab I.

BAB V

PEMBANGUNAN SISTEM

5.1 PENGENALAN

Perisian aplikasi merupakan program yang mengarahkan komputer untuk membuat apa yang kita perlukan. Misalnya ia digunakan untuk memproses perkataan, membentuk pangkalan data, membentuk dokumen yang menarik, membuat carta dan lain-lain lagi seperti *Microsoft Access*. Perisian aplikasi bertujuan menjayakan tugas-tugas pengguna akhir. Pembangunan sistem atau aplikasi selalunya dibuat mengikut permintaan klien tentang sistem yang bagaimana mereka mahukan. Perbincangan akan dibuat terlebih dahulu dengan klien mengenai reka bentuk sistem dan juga keperluan lain yang diperlukan untuk pembangunan sistem yang klien mahukan.

5.2 PEMBANGUNAN SISTEM

Pembangunan sistem merangkumi dua aspek iaitu kod pengaturcaraan dalam menghasilkan sistem dan juga antara muka untuk pengguna sistem ini. Objektif utama yang perlu dititikberatkan ialah pengaplikasian metod Topeng R-CNN (*Mask R-CNN*) dalam pengecaman objek iaitu udang. Metod ini digunakan semasa melakukan latihan sistem dan juga pengujian sistem terhadap gambar yang dimasukkan. Antara muka untuk pengguna sistem ini mengutamakan paparan jumlah bilangan udang yang dikesan setelah gambar udang dimasukkan. Pada waktu yang sama, set data diperlukan untuk menguji sistem ini. Set data yang dikumpul merangkumi tapak udang iaitu Manjung Aquabest Hatchery yang berada di Lumut, Perak. Berikut merupakan fasa-fasa pembangunan sistem:

1. Fasa Perolehan Perkakasan
2. Fasa Perolehan Set Data
3. Fasa Latihan Model
4. Fasa Implementasi

5.2.1 Fasa Perolehan Perkakasan

Sistem ini akan diuji di tapak udang iaitu di Manjung Aquabest Hatchery yang berada di Lumut, Perak. Perbincangan telah dilakukan untuk membincangkan keperluan sistem bersama pengusaha tapak udang tersebut untuk memberi pendapat. Hasil perbincangan tersebut, beberapa keperluan sistem dapat dikenal pasti dan perkara ini dijadikan panduan objektif kajian ini dilaksanakan.

Fasa perolehan perkakasan merangkumi pembelian perkakasan-perkakasan yang berkaitan untuk memastikan sistem ini berjalan dengan baik seperti kamera DSLR, kamera bermodul USB, lampu LED, komputer dan lain-lain. Resolusi kamera yang berkualiti tinggi diperlukan untuk memastikan perolehan set data berkualiti dan set data tersebut boleh digunakan untuk sistem ini dan dapat memberikan keputusan pengecaman yang terbaik. Seterusnya, sudut kamera juga memainkan peranan penting. Sudut dari atas atau sudut yang membolehkan mengeluarkan ciri-ciri udang untuk proses pengecaman objek dengan baik adalah seperti Rajah 5.1 yang menunjuk di bawah.



Rajah 5.1 : Gambar yang diambil menggunakan kamera DSLR

5.2.2 Fasa Perolehan Set Data

Fasa ini merupakan fasa untuk perolehan set data yang diperlukan untuk melakukan latihan sistem iaitu dalam fasa latihan model. Set data diperolehi dengan menggunakan kamera dan gambar diambil di tapak udang tersebut. Jumlah gambar yang telah diambil adalah sebanyak 150 keping. Antara perkara yang dititikberatkan dalam perolehan set data ialah jumlah udang dan kepadatan udang. Kepadatan udang akan terbahagi kepada tiga jenis iaitu kurang padat, sederhana padat dan padat. Kepadatan udang adalah seperti berikut:-

- 1) Kurang Padat (Rujuk Rajah 5.2)



Rajah 5.2 : Gambar yang menunjukkan udang dalam keadaan kurang padat

- 2) Sederhana Padat (Rujuk Rajah 5.3)



Rajah 5.3 : Gambar yang menunjukkan udang dalam keadaan sederhana padat

3) Padat (Rujuk Rajah 5.4)



Rajah 5.4 : Gambar yang menunjukkan udang dalam keadaan padat

5.2.3 Fasa Latihan Model

Dalam fasa latihan model, perisian yang digunakan adalah *Google Colab*, *VGG Image Annotator* dan bahasa pengaturcaraan digunakan adalah *Python*. Sebanyak 100 gambar dalam folder *train* akan digunakan dalam fasa ini. Dalam fasa latihan model, *Google Colab* dipilih untuk melatih model disebabkan oleh penghasilan model akan bergantung kepada RAM dan GPU. *Google Colab* menyediakan ruang RAM dan GPU yang sangat mencukupi untuk menghasilkan model. Hasil daripada fasa ini adalah model pengiraan bilangan udang.

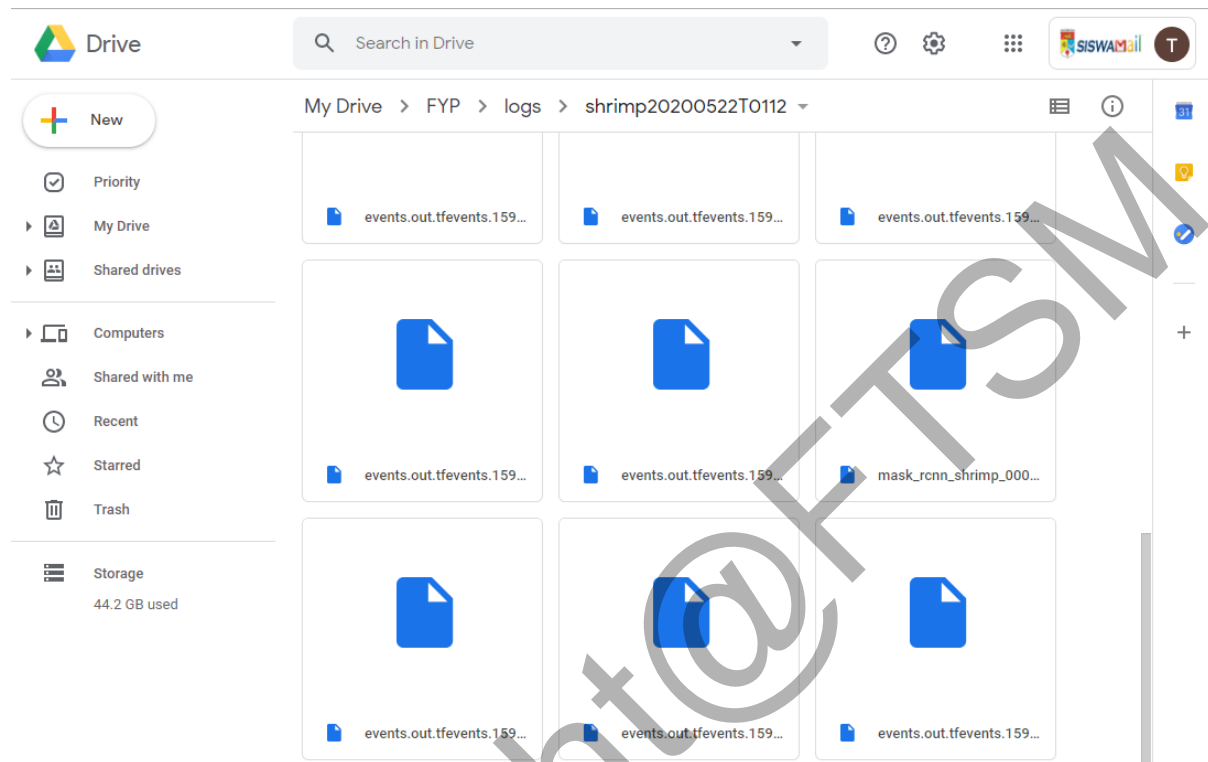
```

1 %pip install -r /content/drive/My Drive/FYP/requirements.txt
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/lib/python3.6/dist-packages (from matplotlib->r /content/drive/My Drive/FYP/requirements.txt)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.6/dist-packages (from matplotlib->r /content/drive/My Drive/FYP/requirements.txt (line 1))
Requirement already satisfied: cyclert>=0.10 in /usr/local/lib/python3.6/dist-packages (from matplotlib->r /content/drive/My Drive/FYP/requirements.txt (line 5))
Requirement already satisfied: PyWavelets>=0.4.0 in /usr/local/lib/python3.6/dist-packages (from scikit-image->r /content/drive/My Drive/FYP/requirements.txt (line 1))
Requirement already satisfied: imageio>=2.3.0 in /usr/local/lib/python3.6/dist-packages (from scikit-image->r /content/drive/My Drive/FYP/requirements.txt (line 1))
Requirement already satisfied: networkx>=2.0 in /usr/local/lib/python3.6/dist-packages (from scikit-image->r /content/drive/My Drive/FYP/requirements.txt (line 1))
Requirement already satisfied: six in /usr/local/lib/python3.6/dist-packages (from h5py->r /content/drive/My Drive/FYP/requirements.txt (line 8)) (1.12.0)
Requirement already satisfied: Shapely in /usr/local/lib/python3.6/dist-packages (from imgaug->r /content/drive/My Drive/FYP/requirements.txt (line 9)) (1.7.0)
Requirement already satisfied: decorator in /usr/local/lib/python3.6/dist-packages (from IPython[all]->r /content/drive/My Drive/FYP/requirements.txt (line 10))
Requirement already satisfied: setuptools>=18.5 in /usr/local/lib/python3.6/dist-packages (from IPython[all]->r /content/drive/My Drive/FYP/requirements.txt (line 10))
Requirement already satisfied: prompt-toolkit<2.0.0,>=1.0.4 in /usr/local/lib/python3.6/dist-packages (from IPython[all]->r /content/drive/My Drive/FYP/requirements.txt (line 10))
Requirement already satisfied: pickleshare in /usr/local/lib/python3.6/dist-packages (from IPython[all]->r /content/drive/My Drive/FYP/requirements.txt (line 10))
Requirement already satisfied: pexpect; sys_platform != "win32" in /usr/local/lib/python3.6/dist-packages (from IPython[all]->r /content/drive/My Drive/FYP/requirements.txt (line 10))
Requirement already satisfied: pygments in /usr/local/lib/python3.6/dist-packages (from IPython[all]->r /content/drive/My Drive/FYP/requirements.txt (line 10))
Requirement already satisfied: simplegeneric<0.8 in /usr/local/lib/python3.6/dist-packages (from IPython[all]->r /content/drive/My Drive/FYP/requirements.txt (line 10))
Requirement already satisfied: traitlets>=4.2 in /usr/local/lib/python3.6/dist-packages (from IPython[all]->r /content/drive/My Drive/FYP/requirements.txt (line 10))
Requirement already satisfied: ipyparallel; extra == "all" in /usr/local/lib/python3.6/dist-packages (from IPython[all]->r /content/drive/My Drive/FYP/requirements.txt (line 10))
Requirement already satisfied: qtconsole; extra == "all" in /usr/local/lib/python3.6/dist-packages (from IPython[all]->r /content/drive/My Drive/FYP/requirements.txt (line 10))
Requirement already satisfied: nbformat; extra == "all" in /usr/local/lib/python3.6/dist-packages (from IPython[all]->r /content/drive/My Drive/FYP/requirements.txt (line 10))
Requirement already satisfied: ipynbutils; extra == "all" in /usr/local/lib/python3.6/dist-packages (from IPython[all]->r /content/drive/My Drive/FYP/requirements.txt (line 10))
Requirement already satisfied: testpath; extra == "all" in /usr/local/lib/python3.6/dist-packages (from IPython[all]->r /content/drive/My Drive/FYP/requirements.txt (line 10))
Requirement already satisfied: nbconvert; extra == "all" in /usr/local/lib/python3.6/dist-packages (from IPython[all]->r /content/drive/My Drive/FYP/requirements.txt (line 10))
Requirement already satisfied: sphinx>=1.3; extra == "all" in /usr/local/lib/python3.6/dist-packages (from IPython[all]->r /content/drive/My Drive/FYP/requirements.txt (line 10))
Requirement already satisfied: nose>=0.10.1; extra == "all" in /usr/local/lib/python3.6/dist-packages (from IPython[all]->r /content/drive/My Drive/FYP/requirements.txt (line 10))
Requirement already satisfied: ipykernel; extra == "all" in /usr/local/lib/python3.6/dist-packages (from IPython[all]->r /content/drive/My Drive/FYP/requirements.txt (line 10))
Requirement already satisfied: notebook; extra == "all" in /usr/local/lib/python3.6/dist-packages (from IPython[all]->r /content/drive/My Drive/FYP/requirements.txt (line 10))

```

Rajah 5.5 : Antara muka bagi *Google Colab*

Rajah 5.6 menunjukkan penghasilan model selepas fasa latihan model. Model tersebut akan disimpan di *Google Drive*. Model tersebut akan digunakan semasa fasa implementasi dalam proses pengiraan yang seterusnya dan ketepatan bagi model tersebut juga akan diuji.



Rajah 5.6 : Penghasilan model selepas fasa latihan model

5.2.4 Fasa Implementasi

Dalam fasa implementasi, model akan digunakan untuk membuat anggaran terhadap imej yang diletakkan dalam folder di *Google Colab*. Model tersebut akan mengesan udang yang terdapat dalam imej dan kotak tersebut akan diwarnakan dengan pelbagai warna. Maklumat berkenaan akan dipaparkan di atas kotak berwarna. Maklumat berkenaan merupakan nama kelas iaitu udang (*shrimp*) dan ketepatan bagi setiap kotak yang telah diwarnakan.

Predictions



Rajah 5.7 : Hasil dan keputusan bagi salah satu imej

Rajah 5.7 menunjukkan salah satu hasil dan keputusan imej dengan menggunakan model tersebut untuk membuat anggaran terhadap imej yang diletakkan dalam folder.

```

X image ID: shrimp.0081.JPG (0) /content/drive/My Drive/FYP/dataset/val/0081.JPG
Processing 1 images
image           shape: (1024, 1024, 3)    min:  0.00000  max:  254.00000  uint8
molded_images   shape: (1, 1024, 1024, 3) min: -123.70000  max:  150.10000  float64
image metas     shape: (1, 14)           min:  0.00000  max:  1024.00000  int64
anchors        shape: (1, 261888, 4)    min: -0.35390  max:  1.29134  float32
Total No of Shrimp Predicted: 75

```

Rajah 5.8 : Maklumat dan bilangan udang terdapat dalam imej (Rujuk Rajah 5.7)

Rajah 5.8 menunjukkan maklumat dan bilangan udang yang dapat digesa dalam imej tersebut. Berdasarkan Rajah 5.7, boleh didapati mempunyai 75 ekor udang dalam imej tersebut.

5.3 KESIMPULAN

Kesimpulannya, bab ini membincangkan implementasi model terhadap pengesanan objek dan pengiraan objek udang. Turut juga membincangkan beberapa fasa-fasa sebelum menjalankan fasa implementasi dapat dijalankan seperti fasa perolehan perkakasan, fasa perolehan set data dan fasa latihan model.

Copyright@FTSM

BAB VI

PERANCANGAN PENGUJIAN

6.1 PENGENALAN

Perancangan Pengujian merupakan aktiviti verifikasi yang dilakukan terhadap komponen atau sistem untuk memastikan ia dibangunkan berdasarkan kepada spesifikasi keperluan dan reka bentuk sistem. Perancangan Pengujian merangkumi pelbagai peringkat ujian sebelum sistem diuji secara komprehensif di dalam fasa pengujian penerimaan. Semasa pengujian ini dilaksanakan, ralat yang dikesan akan diperbetulkan dan unit/komponen/modul yang berkaitan akan diuji semula sehingga ralat berjaya diperbaiki.

6.2 REKA BENTUK UNTUK PROTOTAIP

Reka bentuk untuk prototaip dibangunkan untuk memudahkan proses pengiraan udang. Melalui prototaip ini, pengguna hanya perlu meletakkan udang dalam bekas dan mengambil gambar untuk menggunakan kamera yang telah diletak dalam prototaip tersebut. Prototaip ini perlu bersambung dengan komputer supaya dapat mengambil gambar. Dalam prototaip tersebut juga akan meletakkan lampu obor (*torch light*) supaya udang dalam bekas dapat dilihat dengan lebih jelas dan mudah digesa.

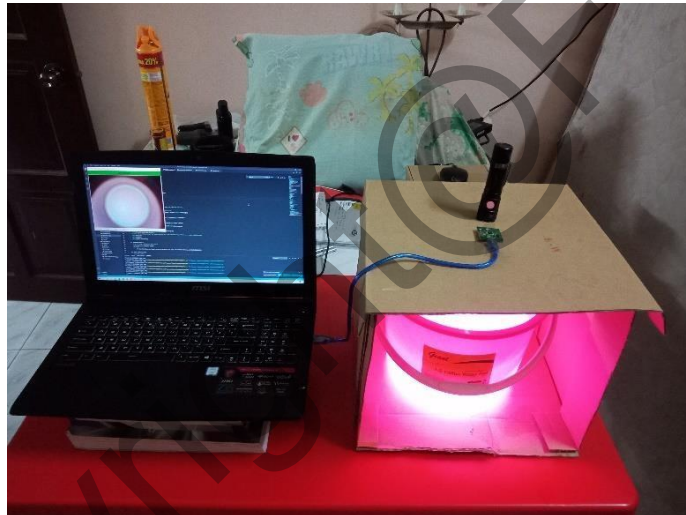


Rajah 6.1 : Reka bentuk untuk prototaip

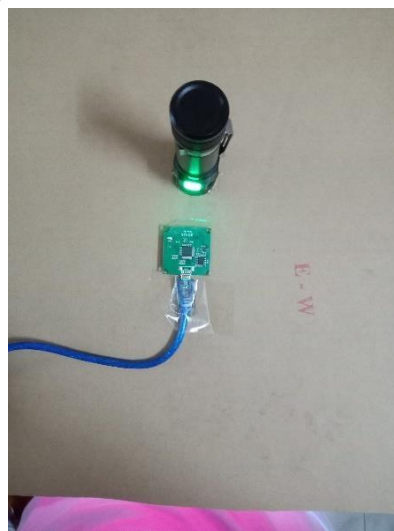
6.3 PROSES PENGAMBILAN GAMBAR

Untuk proses pengambilan gambar, satu antara muka akan dibangunkan untuk kemudahan mengambil gambar. Untuk pixel bagi pengambilan gambar adalah 640 pixel untuk lebar dan 480 pixel untuk ketinggian. Kamera yang digunakan adalah Arducam USB Camera Module 8MP Sony IMX219. Lampu obor yang digunakan adalah seperti biasa. Gambar yang diambil akan dalam format .jpg. Ketinggian tong air yang digunakan adalah 20 cm dan diameter sebanyak 25.5cm. Lampu obor akan dilaraskan sampai semua cahaya dapat menutup permukaan air dalam tong air tersebut. Berikut adalah peralatan yang digunakan dalam prototaip:

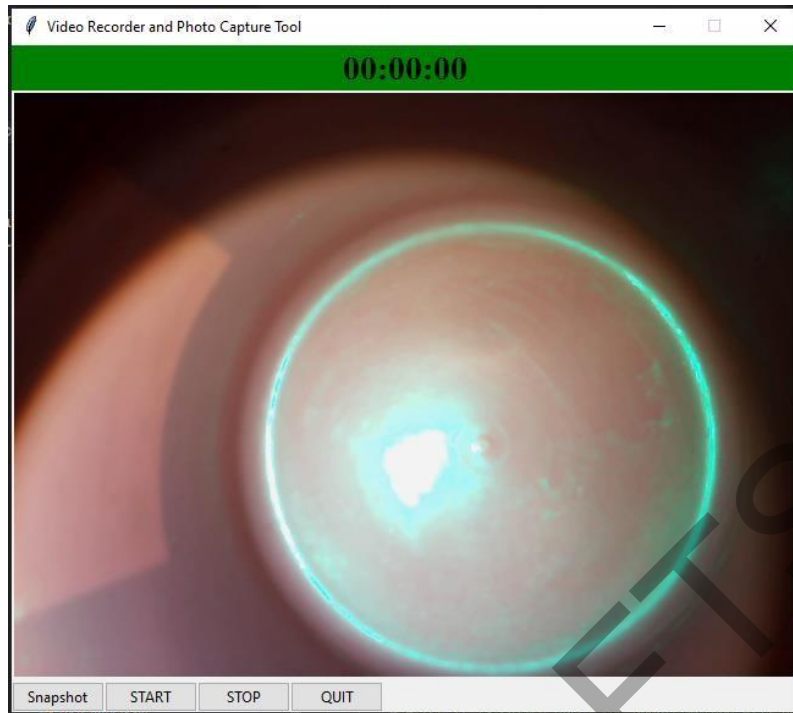
1. Kotak
2. Tong air
3. Arducam USB Camera Module 8MP Sony IMX219
4. Lampu obor (torch light)



Rajah 6.2 : Persediaan bagi proses pengambilan gambar



Rajah 6.3 : Pandangan dari atas bagi prototaip



Rajah 6.4 : Antara muka untuk proses pengambilan gambar

Langkah 1 : Berhubung kamera Arducam USB Camera Module 8MP Sony IMX219 ke komputer.

Langkah 2 : Buka antara muka bagi pengambilan gambar.

Langkah 3 : Letakkan udang yang ingin dikira.

Langkah 4 : Membuka lampu obor.

Langkah 5 : Melaraskan kecahayaan lampu obor sehingga menutup permukaan air.

Langkah 6 : Proses pengiraan udang bermula.

6.4 DATA UJIAN

Dalam folder bagi set data mempunyai dua folder iaitu *train* dan *val*. Data ujian adalah bagi menilai ketepatan berkenaan dengan model yang telah dilatih dalam fasa latihan model. Dalam folder *val* mempunyai 20 gambar dipilih dengan secara rawak untuk sebagai data ujian manakala bagi folder *train* mempunyai 100 gambar. Data bagi ujian adalah dalam format *.jpg* atau *.jpeg*. Saiz atau resolusi untuk imej telah bertukar daripada 6000 piksel bagi lebar dan 4000 piksel bagi ketinggian kepada 960 piksel bagi lebar dan 640 piksel bagi ketinggian. Saiz atau resolusi yang kecil yang dimuat naik dapat mempercepatkan proses pengiraan udang.



Rajah 6.5 : Sebahagian gambar yang simpan dalam folder *val*

6.5 PENETAPAN PARAMETER ANALISIS UNTUK MODEL LATIHAN

Proses menetapkan parameter memerlukan kepakaran dan percubaan yang banyak kali serta kemungkinan akan berlaku kesilapan. Tidak mempunyai cara yang mudah untuk menetapkan parameter khususnya, *learning rate*, *batch size*, *learning momentum*, dan *weight decay*. Sumber untuk model pembelajaran mendalam (*Deep Learning*) penuh dengan parameter dan mencari konfigurasi yang terbaik untuk parameter seperti ini merupakan salah satu cabaran yang besar. Parameter ini bertindak sebagai tombol yang boleh diubah semasa latihan model. Untuk memberikan hasil yang terbaik, nilai optimum parameter perlu dicari. Jadual 6.1 menunjukkan parameter analisis yang asal untuk model latihan selepas nilai optimum parameter dicari.

	Mask RCNN
BACKBONE	Resnet101
BACKBONE_STRIDES	[4, 8, 16, 32, 64]
BATCH_SIZE	2
BBOX_STD_DEV	[0.1 0.1 0.2 0.2]
COMPUTE_BACKBONE_SHAPE	None
DETECTION_MAX_INSTANCES	100
DETECTION_MIN_CONFIDENCE	0.7
DETECTION_NMS_THRESHOLD	0.3
FPN_CLASSIF_FC_LAYERS_SIZE	1024
GPU_COUNT	1
GRADIENT_CLIP_NORM	5.0
IMAGES_PER_GPU	2
IMAGE_CHANNEL_COUNT	3
IMAGE_MAX_DIM	1024
IMAGE_META_SIZE	14
IMAGE_MIN_DIM	800
IMAGE_MIN_SCALE	0
IMAGE_RESIZE_MODE	Square
IMAGE_SHAPE	[1024 1024 3]
LEARNING_MOMENTUM	0.9
LEARNING_RATE	0.001
LOSS_WEIGHTS	{'rpn_class_loss': 1.0, 'rpn_bbox_loss': 1.0, 'mrcnn_class_loss': 1.0, 'mrcnn_bbox_loss': 1.0, 'mrcnn_mask_loss': 1.0}
MASK_POOL_SIZE	14
MASK_SHAPE	[28, 28]
MAX_GT_INSTANCES	100
MEAN_PIXEL	[123.7 116.8 103.9]
MINI_MASK_SHAPE	(56, 56)
NAME	shrimp
NUM_CLASSES	2
POOL_SIZE	7
POST_NMS_ROIS_INFERENCE	1000
POST_NMS_ROIS_TRAINING	2000
PRE_NMS_LIMIT	6000
ROI_POSITIVE_RATIO	0.33
RPN_ANCHOR RATIOS	[0.5, 1, 2]
RPN_ANCHOR_SCALES	(32, 64, 128, 256, 512)
RPN_ANCHOR_STRIDE	1
RPN_BBOX_STD_DEV	[0.1 0.1 0.2 0.2]
RPN_NMS_THRESHOLD	0.7
RPN_TRAIN_ANCHORS_PER_IMAGE	256
STEPS_PER_EPOCH	50
TOP_DOWN_PYRAMID_SIZE	256
TRAIN_BN	False
TRAIN_ROIS_PER_IMAGE	200

USE_MINI_MASK	True
USE_RPN_ROIS	True
VALIDATION_STEPS	50
WEIGHT_DECAY	0.0001

Jadual 6.1 : Parameter analisis yang asal

6.6 PENGENALAN DENGAN *HYPERPARAMETER*

Dalam setiap algoritma bagi pembelajaran bukan mesin, pembelajaran mesin dan pembelajaran mendalam mempunyai *hyperparameter*. *Hyperparameter* ini akan menyebabkan prestasi model tersebut dalam proses membuat anggaran. *Hyperparameter* adalah sifat yang mengatur keseluruhan semasa fasa latihan model. Ia merangkumi pemboleh ubah yang menentukan bagaimana model dilatih seperti *learning rate*. *Hyperparameter* bagi model ditetapkan sebelum fasa latihan model dijalankan.

6.7 *HYPERPARAMETER* UNTUK DIUJI

Selepas fasa latihan model dengan menggunakan parameter analisis yang asal (Rujuk Jadual), beberapa *hyperparameter* telah diubah berdasarkan ujian terhadap model tersebut. Parameter adalah seperti berikut :

- 1) *DETECTION_MAX_INSTANCE*
- 2) *MAX_GT_INSTANCE*
- 3) *RPN_NMS_THRESHOLD*
- 4) *RPN_TRAIN_ANCHORS_PER_IMAGE*
- 5) *TRAIN_ROIS_PER_IMAGE*
- 6) *VALIDATION_STEPS*
- 7) *STEPS_PER_EPOCH* DAN *NUMBERS OF EPOCH*
- 8) *REGULARIZATION*
- 9) *OPTIMIZERS*
- 10) *LEARNING RATE*
- 11) *BATCH SIZE*
- 12) *LEARNING MOMENTUM*
- 13) *WEIGHT DECAY*

6.7.1 *DETECTION_MAX_INSTANCE*

Parameter untuk *DETECTION_MAX_INSTANCE* merupakan bilangan maksimum untuk bilangan tekaan dalam setiap imej. Bilangan maksimum bagi bilangan tekaan adalah sebanyak 213 ekor udang (Rujuk Rajah 6.6).



Rajah 6.6 : Gambar yang mempunyai bilangan maksimum bagi bilangan tekaan

6.7.2 *MAX_GT_INSTANCE*

Parameter untuk *MAX_GT_INSTANCE* merupakan bilangan maksimum untuk bilangan sebenar dalam semua imej. Bilangan maksimum bagi bilangan sebenar adalah sebanyak 256 ekor udang (Rujuk Rajah 6.7).



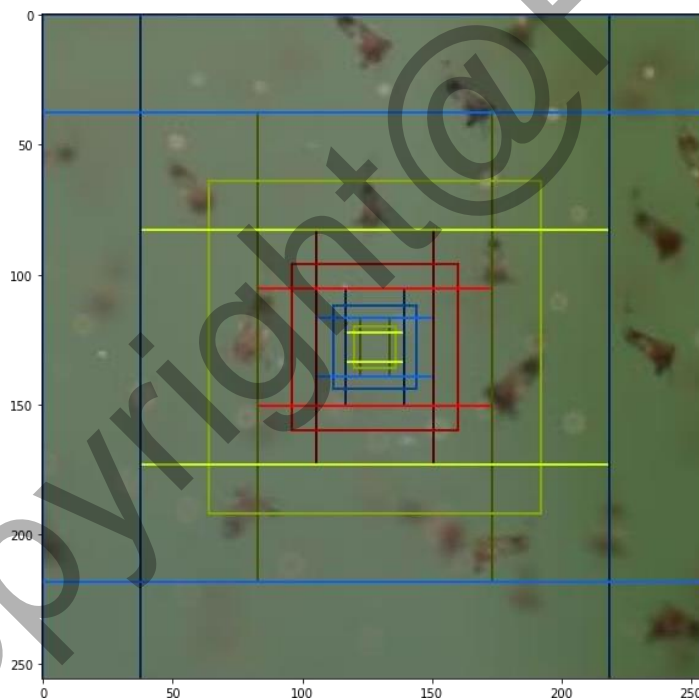
Rajah 6.7 : Gambar yang mempunyai bilangan maksimum bagi bilangan sebenar

6.7.3 *RPN_NMS_THRESHOLD*

Parameter untuk *RPN_NMS_THRESHOLD* juga bergantung kepada nilai bagi parameter *RPN_TRAIN_ANCHORS_PER_IMAGE*. Parameter bagi *RPN_NMS_THRESHOLD* akan ditukar kepada 0.8 daripada 0.7. Parameter ini akan memilih *Region Proposal Network* (RPN) yang mematuhi syarat iaitu bernilai lebih daripada 0.8. Selain daripada nilai 0.8 atau kurang daripada nilai 0.8 tidak akan dipilih sebagai *Region Proposal Network* (RPN).

6.7.4 *RPN_TRAIN_ANCHORS_PER_IMAGE*

Parameter untuk *RPN_TRAIN_ANCHORS_PER_IMAGE* merupakan beberapa kotak akan dijana di setiap imej bagi setiap tahap (Rujuk Rajah 6.8). Dengan menambahkan bilangan bagi parameter ini akan bertambah lebih banyak kotak dijanakan di imej tersebut. Kotak ini akan menangkap setiap piksel dalam imej dengan lebih terperinci. Parameter untuk *RPN_TRAIN_ANCHORS_PER_IMAGE* akan ditukar kepada 512 daripada 256.

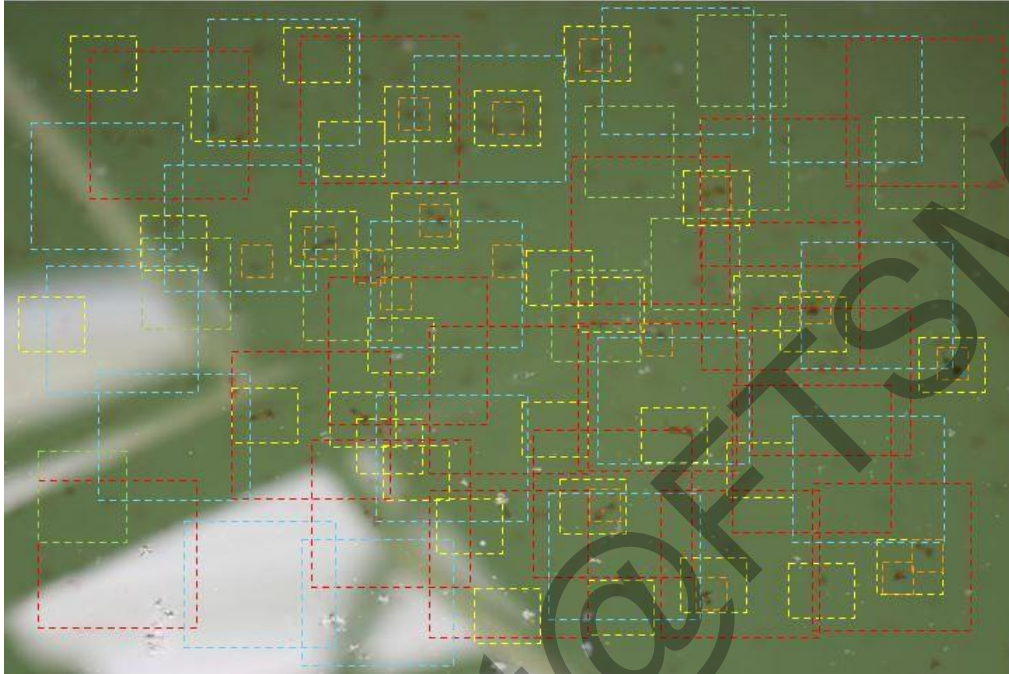


Rajah 6.8 : Kotak yang dijana di salah satu imej bagi setiap tahap

6.7.5 *TRAIN_ROIS_PER_IMAGE*

RoIs merupakan perkataan *Regions of Interests*. Dalam algoritma Topeng R-CNN (*Mask R-CNN*) menggunakan pengeks trak untuk mengekstrak ciri imej. Kemudian, ia menggunakan Rangkaian Cadangan Wilayah CNN (*CNN Region Proposal Network*) untuk membuat RoIs. Penyatuan RoI akan digunakan untuk melengkapkan ke dimensi yang tetap. Kemudian dimasukkan ke lapisan yang terhubung sepenuhnya untuk membuat klasifikasi (*classification*)

dan ramalan kotak sempadan (*boundary box prediction*). Parameter untuk TRAIN_ROIS_PER_IMAGE akan ditukar kepada 300 daripada 200. Parameter ini merupakan bilangan bagi RoIs yang akan dijana dalam satu imej.



Rajah 6.9 : Contoh ROIS yang dijanakan dalam gambar

6.7.6 VALIDATION_STEPS

Parameter *VALIDATION_STEPS* adalah merupakan berapa langkah akan dijalankan untuk mengadakan ujian terhadap *validation* set data. Dalam folder val, mempunyai 20 imej, dan telah meningkat kepada 200 langkah daripada 50 langkah. Langkah *validation* akan dijalankan untuk setiap imej bagi setiap epoch semasa fasa latihan model. Parameter ini akan mempengaruhi *validation loss* (*val loss*).

6.7.7 STEPS_PER_EPOCH Dan NUMBERS OF EPOCH

Parameter *STEPS_PER_EPOCH* digunakan untuk menghasilkan keseluruhan set data dengan memanggil penjana data dengan mengikut bilangan yang telah set dalam parameter. Parameter *NUMBERS OF EPOCH* adalah berapa kali model perlu dilatih untuk keseluruhan set data. Parameter *STEPS_PER_EPOCH* akan ditukar kepada 100 daripada 50. Parameter *NUMBERS OF EPOCH* akan kekal kepada 50.

6.7.8 REGULARIZATION

Peranan regularisasi (*regularization*) adalah memodifikasi model pembelajaran mendalam agar dapat berfungsi dengan baik dengan set data. Secara khususnya, regularisasi menumpukan pada mengurangkan kesalahan ujian atau generalisasi (*generalization*) tanpa mempengaruhi kesalahan dalam fasa latihan model yang peringkat awal.

L1 dan L2 adalah jenis regularisasi (*regularization*) yang paling biasa. Regularisasi (*Regularization*) akan mengemas kini fungsi kos umum (*general cost function*) dengan menambahkan istilah lain yang dikenali sebagai istilah regularisasi (*regularization*).

$$J(\theta_0, \theta_1) = \frac{1}{2n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^n \theta_j^2$$

Formula 6.1 : Formula untuk Fungsi Kos (*Cost Function*)

Dalam L1, formula yang digunakan adalah seperti berikut :

$$J(\theta_0, \theta_1) = \frac{1}{2n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^n |\theta_j|$$

Formula 6.2 : Formula untuk Fungsi Kos (*Cost Function*) bagi L1

Dalam L1, nilai mutlak pemberat (*absolute value of weights*) akan dihukum. Tidak seperti dalam L2, nilai mutlak pemberat (*absolute value of weights*) boleh dikurangkan menjadi 0. Oleh itu, L1 sangat berguna ketika dalam proses memampatkan model.

Dalam L2, formula yang digunakan adalah seperti berikut :

$$J(\theta_0, \theta_1) = \frac{1}{2n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \frac{\lambda}{2} \sum_{j=1}^n \theta_j^2$$

Formula 6.3 : Formula untuk Fungsi Kos (*Cost Function*) bagi L2

Dalam L2, *lambda* (λ) adalah parameter untuk regularisasi (*regularization*). Ia adalah *hyperparameter* yang dioptimumkan untuk hasil yang lebih baik. L2 juga dikenali sebagai *weight decay* kerana L2 memaksa *weight decay* menurun ke 0 tetapi tidak semestinya 0.

Parameter untuk regularisasi (*Regularization*) yang digunakan adalah L1 dan L2. Perbezaan antara L1 dan L2 akan diuji dari segi prestasi semasa dalam fasa latihan model.

6.7.9 OPTIMIZERS

Terdapat banyak jenis pengoptimuman dalam pembelajaran mendalam tetapi paling relevan yang difokuskan kepada pengurangan fungsi kos terhadap model. Di antara algoritma pengoptimuman dalam pembelajaran mendalam, *Stochastic Gradient Descent* (SGD) telah menjadi variasi yang paling popular dalam kalangan penggunaan di pembelajaran mendalam. Umumnya, terdapat banyak variasi SGD seperti SGD dengan Momentum yang berfungsi lebih baik dalam algoritma pembelajaran mendalam tertentu. Pengoptimuman dalam model pembelajaran mendalam adalah benar-benar kombinasi antara teknik regularisasi dengan teknik pengoptimuman, Parameter untuk pengoptimuman yang digunakan adalah *Stochastic Gradient Descent* (SGD).

6.7.10 LEARNING RATE

Kadar pembelajaran (*learning rate*) adalah jumlah pemberat (*weights*) yang diperbaharui semasa fasa latihan model. Secara khususnya, kadar pembelajaran (*learning rate*) adalah *hyperparameter* yang dapat dikonfigurasi akan digunakan dalam fasa latihan model yang mempunyai nilai positif kecil, selalunya nilai untuk kadar pembelajaran (*learning rate*) antara 0.0 dan 1.0. Parameter kadar pembelajaran akan diuji adalah 0.010, 0.0010 dan 0.0001.

6.7.11 BATCH SIZE

Parameter *BATCH SIZE* merupakan parameter untuk menentukan berapa imej akan dilatih semasa fasa latihan model. Proses melatih model adalah dilatih di *Google Colab*. Dengan RAM yang sedia ada di *Google Colab* mempunyai 12.72 GB. Maksimum kapasiti bagi *Batch Size* untuk melatih model latihan adalah 1 atau 2. Ia bergantung kepada saiz imej yang digunakan. Untuk *Batch Size* bernilai 2 hanya boleh melatih 2 imej bersaiz 1024 pixel x 1024 pixel pada masa yang sama.

6.7.12 LEARNING MOMENTUM

Teknik yang sangat popular yang digunakan bersama dengan SGD disebut *Momentum*. Daripada hanya menggunakan kecerunan langkah semasa dalam proses pencarian, *momentum* juga mengumpulkan kecerunan langkah-langkah masa lalu untuk menentukan arah yang harus dilalui. *Momentum* juga disebut sebagai penurunan kecenderungan stokastik dan menambah inersia (*inertia*) pada prosedur pembaruan. Ia akan menyebabkan banyak kemas kini terhadap satu arah supaya arah tersebut dapat mengarah ke destinasi yang ditetapkan. Parameter untuk *Learning Momentum* akan diuji adalah 0, 0.5 dan 0.9.

6.7.13 WEIGHT DECAY

Weight Decay, biasanya disebut sebagai regularisasi (*Regularization*) L2, mungkin merupakan teknik yang paling banyak digunakan untuk mengatur parametrik dalam model pembelajaran mesin. *Weight Decay* adalah istilah tambahan yang menyebabkan nilai pemberat (*weight*) menurun secara eksponensial menjadi nilai 0. Parameter untuk *Weight Decay* akan diuji adalah 0.1, 0.01 dan 0.001.

6.8 KESIMPULAN

Kesimpulannya, perancangan pengujian mestilah menjalankan dengan pelbagai aspek ujian seperti pengujian dilakukan secara berperingkat bermula daripada pengujian parameter. Dengan melalui pengujian, prestasi bagi model dapat sentiasa dalam keadaan yang baik dan dapat memastikan hasil keputusan yang didapatkan adalah tepat.

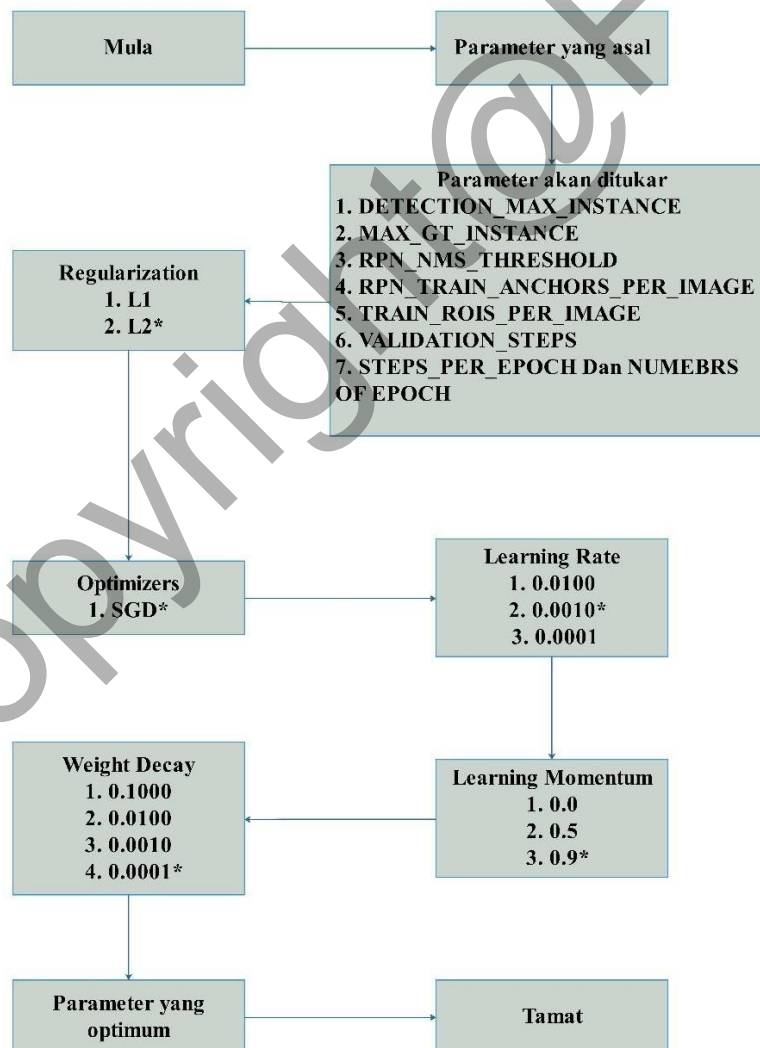
BAB VII

HASIL KAJIAN

7.1 PENGENALAN

Hasil kajian merupakan aktiviti perbincangan yang dilakukan terhadap perancangan pengujian. Beberapa perbincangan dan hasil kajian akan dibincangkan termasuklah rangka kerja untuk memilih *hyperparameter* yang optimum semasa fasa latihan model, beberapa kaedah untuk menguji prestasi terhadap model latihan juga akan dibincangkan dalam bab ini. Hasil daripada pengujian model baharu telah dianalisis dalam bab ini.

7.2 CARTA ALIRAN UNTUK MEMILIH *HYPERPARAMETER* YANG OPTIMUM



* Parameter yang asal

Rajah 7.1 : Carta aliran untuk memilih *hyperparameter* yang optimum

Merujuk kepada Rajah 7.1, pada permulaan, parameter yang asal akan digunakan dalam fasa latihan model. Selepas mendapat keputusan, beberapa parameter akan ditukar untuk mengikut situasi supaya dapat memenuhi set data yang digunakan. Parameter yang ditukar seperti DETECTION_MAX_INSTANCE dan MAX_GT_INSTANCE. Parameter tersebut akan bertukar kepada 400 kerana maksimum bilangan sebenar udang dalam set data adalah sebanyak 256 ekor. Pada langkah yang seterusnya, penukaran atas regularisasi juga akan diuji. Pengujian atas regularisasi adalah antara L1 dan L2. Regularisasi yang mempunyai *loss* yang paling rendah akan dikekalkan. Hanya satu *optimizer* yang akan digunakan iaitu SGD (*Stochastic Gradient Descent*).

Regularisasi yang mempunyai *loss* paling rendah dengan *optimizer* SGD akan dibawa ke langkah seterusnya, iaitu penukaran terhadap kadar pembelajaran (*learning rate*). Nilai bagi kadar pembelajaran (*learning rate*) akan diuji adalah 0.0100, 0.0010 dan 0.0001. Kadar pembelajaran (*learning rate*) yang mempunyai *loss* yang paling rendah akan dikekalkan dan dibawa ke langkah seterusnya, iaitu penukaran terhadap *learning momentum*. Nilai bagi *learning momentum* akan diuji adalah 0.0, 0.5 dan 0.9. Nilai *learning momentum* yang mempunyai *loss* yang paling rendah juga akan dikekalkan dan dibawa ke langkah seterusnya, iaitu penukaran terhadap *weight decay*. Nilai bagi *weight decay* akan diuji adalah 0.100, 0.010 dan 0.001.

Pada langkah yang terakhir, parameter yang optimum akan dipilih dan model tersebut akan digunakan dalam fasa yang seterusnya iaitu fasa implementasi. Prestasi terhadap model juga akan diuji dari segi ketepatan dan sebagainya.

7.3 PARAMETER ANALISIS YANG OPTIMUM UNTUK MODEL LATIHAN

Berikut adalah jadual bagi parameter analisis yang optimum untuk model latihan yang mempunyai *loss* yang paling rendah.

	Mask R-CNN	Improved Mask R-CNN
REGULARIZATION	L2	L1
BACKBONE	Resnet101	Resnet101
BACKBONE_STRIDES	[4, 8, 16, 32, 64]	[4, 8, 16, 32, 64]
BATCH_SIZE	2	2
BBOX_STD_DEV	[0.1 0.1 0.2 0.2]	[0.1 0.1 0.2 0.2]
COMPUTE_BACKBONE_SHAPE	None	None

DETECTION_MAX_INSTANCE	100	400
DETECTION_MIN_CONFIDENCE	0.7	0.7
DETECTION_NMS_THRESHOLD	0.3	0.3
FPN_CLASSIF_FC_LAYERS_SIZE	1024	1024
GPU_COUNT	1	1
GRADIENT_CLIP_NORM	5.0	5.0
IMAGES_PER_GPU	2	2
IMAGE_CHANNEL_COUNT	3	3
IMAGE_MAX_DIM	1024	1024
IMAGE_META_SIZE	14	14
IMAGE_MIN_DIM	800	800
IMAGE_MIN_SCALE	0	0
IMAGE_RESIZE_MODE	Square	Square
IMAGE_SHAPE	[1024 1024 3]	[1024 1024 3]
LEARNING_MOMENTUM	0.9	0.9
LEARNING_RATE	0.001	0.01
LOSS_WEIGHTS	{'rpn_class_loss': 1.0, 'rpn_bbox_loss': 1.0, 'mrcnn_class_loss': 1.0, 'mrcnn_bbox_loss': 1.0, 'mrcnn_mask_loss': 1.0}	{'rpn_class_loss': 1.0, 'rpn_bbox_loss': 1.0, 'mrcnn_class_loss': 1.0, 'mrcnn_bbox_loss': 1.0, 'mrcnn_mask_loss': 1.0}
MASK_POOL_SIZE	14	14
MASK_SHAPE	[28, 28]	[28, 28]
MAX_GT_INSTANCES	100	400
MEAN_PIXEL	[123.7 116.8 103.9]	[123.7 116.8 103.9]
MINI_MASK_SHAPE	(56, 56)	(56, 56)
NAME	shrimp	shrimp
NUM_CLASSES	2	2
POOL_SIZE	7	7
POST_NMS_ROIS_INFERENCE	1000	1000
POST_NMS_ROIS_TRAINING	2000	2000
PRE_NMS_LIMIT	6000	6000
ROI_POSITIVE_RATIO	0.33	0.33
RPN_ANCHOR_RATIOS	[0.5, 1, 2]	[0.5, 1, 2]
RPN_ANCHOR_SCALES	(32, 64, 128, 256, 512)	(32, 64, 128, 256, 512)
RPN_ANCHOR_STRIDE	1	1

RPN_BBOX_STD_DEV	[0.1 0.1 0.2 0.2]	[0.1 0.1 0.2 0.2]
RPN_NMS_THRESHOLD	0.7	0.8
RPN_TRAIN_ANCHORS_PER_IMAGE	256	512
STEPS_PER_EPOCH	50	100
TOP_DOWN_PYRAMID_SIZE	256	256
TRAIN_BN	False	False
TRAIN_ROIS_PER_IMAGE	200	300
USE_MINI_MASK	True	True
USE_RPN_ROIS	True	True
VALIDATION_STEPS	50	200
WEIGHT_DECAY	0.0001	0.001

Jadual 7.1 : Parameter analisis yang optimum untuk model latihan

7.4 HASIL KEPUTUSAN PENGUJIAN DAN PERBINCANGAN

Dalam bahagian ini, prestasi model latihan akan menguji melalui beberapa kaedah seperti *Loss Function*, *Mean Average Precisions (mAP)*, *Precisions*, *Recall*.

7.4.1 Loss Function

Dalam *Loss Function* bagi banyak tugas (*multi-task*) Topeng R-CNN (*Mask R-CNN*) menggabungkan nilai kehilangan (*loss*) bagi klasifikasi (*classification*), penyetempatan (*localization*) dan segmentasi (*segmentation*), untuk $\square_{\square\square\square}$ dan $\square_{\square\square\square}$ adalah sama dengan algoritma seperti di *Faster R-CNN*. Formula adalah seperti berikut :

$$\square = \square_{\square\square\square} + \square_{\square\square\square} + \square_{\square\square\square\square}$$

Formula 7.1 : Formula untuk banyak tugas (*multi-task*) Topeng R-CNN (*Mask R-CNN*)

Untuk menguji prestasi terhadap model, salah satu kaedah adalah menguji dari segi *Loss Function*. Model yang mempunyai *Loss Function* yang paling rendah merupakan model tersebut dapat mengesan objek atau mengklasifikasikan objek dengan baik.

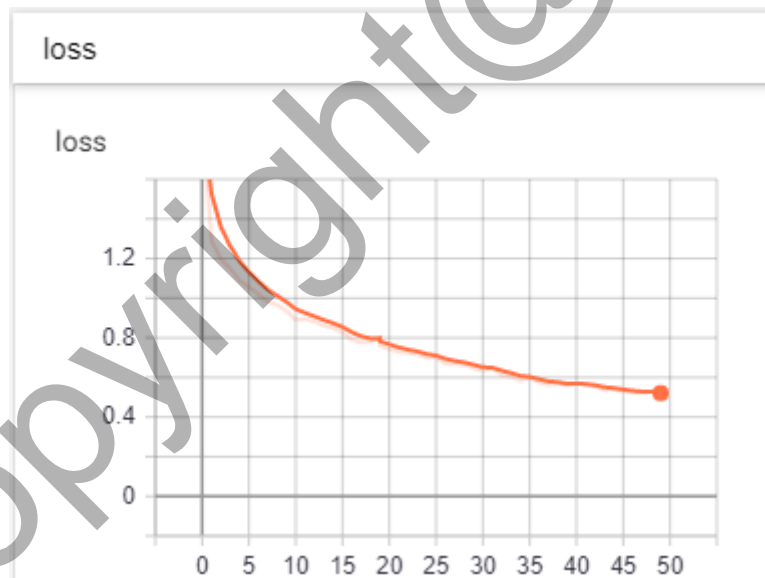
Dalam fasa latihan model, terdapat 5 subjenis *Loss*, iaitu *rpn_class_loss*, *rpn_bbox_loss*, *mrcnn_bbox_loss*, *mrcnn_class_loss* dan *mrcnn_mask_loss*. Untuk jumlah keseluruhan bagi *Loss*, 5 subjenis *Loss* akan ditambahkan. Pengiraan formula *Loss* tersebut akan digunakan dalam set data *train* dan juga set data *val*. Formula untuk jumlah keseluruhan bagi *Loss* adalah seperti berikut :

$$\begin{aligned}
 \square\square\square\square &= \square\square\square\square\square\square\square\square\square\square + \square\square\square\square\square\square\square\square\square\square + \square\square\square\square\square\square\square\square\square\square \\
 &+ \square\square\square\square\square\square\square\square\square\square + \square\square\square\square\square\square\square\square\square\square
 \end{aligned}$$

Formula 7.2 : Formula untuk jumlah keseluruhan bagi *Loss*

Berikut adalah definisi untuk 5 subjenis *Loss*:

- 1) *rpn_class_loss* : Sejauh mana Rangkaian Cadangan Wilayah (*Region Proposal Network - RPN*) memisahkan latar belakang dengan objek.
- 2) *rpn_bbox_loss* : Sejauh mana Rangkaian Cadangan Wilayah (*Region Proposal Network - RPN*) menempatkan objek.
- 3) *mrcnn_bbox_loss* : Sejauh mana Topeng R-CNN (*Mask R-CNN*) menempatkan objek.
- 4) *mrcnn_class_loss* : Sejauh mana Topeng R-CNN (*Mask R-CNN*) mengenali setiap kelas objek.
- 5) *mrcnn_mask_loss* : Sejauh mana Topeng R-CNN (*Mask R-CNN*) dalam segmentasi objek.



Rajah 7.2 : Salah satu *Loss* untuk set data *train*



Rajah 7.3 : Salah satu *Loss* untuk set data *val*

7.4.2 Mean Average Precisions (mAP)

Perkataan mAP (*Mean Average Precisions*) adalah purata bagi AP (*Average Precisions*). Dalam beberapa konteks, AP (*Average Precisions*) akan dikira bagi setiap kelas dan purata bagi setiap AP (*Average Precisions*) akan dikira dan ia adalah mAP (*Mean Average Precisions*). Tetapi dalam beberapa konteks, AP (*Average Precisions*) dan mAP (*Mean Average Precisions*) dianggap sebagai sama. AP (*Average Precisions*) adalah metrik yang popular dalam mengukur ketepatan bagi pengesanan objek seperti *Faster R-CNN*. Berikut adalah keputusan bagi set data *train* dan *val*.

	Backbone	Train		Val	
		mAP (AP _{0.50})	AP _{0.75}	mAP (AP _{0.50})	AP _{0.75}
Mask R-CNN	Resnet101	90.23%	65.85%	95.83%	72.77%
Improved Mask R-CNN	Resnet101	99.00%	96.35%	99.70%	98.50%

Jadual 7.2 : Keputusan mAP bagi set data *train* dan *val*

7.4.3 Precisions dan Recall

Bagi *Precisions* adalah untuk mengukur ketepatan bagi model tersebut seberapa tepat dalam membuat ramalan, iaitu berapa peratusan bagi ramalan yang telah betul. Bagi *Recall* adalah ukuran seberapa baik bagi model tersebut menjumpai semua positif. Berikut adalah formula untuk pengiraan bagi *Precisions* dan *Recall*.

$$\frac{\sum_{i=1}^n \frac{1}{n} (x_i - \bar{x})^2}{n} = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n^2}$$

Formula 7.5 : Formula untuk pengiraan bagi peratusan ketepatan

$$\text{Peratusan Kesilapan} = 100\% - \text{Peratusan Ketepatan}$$

Formula 7.6 : Formula untuk pengiraan bagi peratusan kesilapan

$$\text{Peratusan Kesilapan} = \frac{\sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}}}{\bar{x}} \times 100\%$$

Formula 7.7 : Formula untuk pengiraan bagi sisihan piawaian

Jenis Kepadatan	Bilangan Sebenar (<i>Ground Truth</i>)	Bilangan Tekaan (<i>Predicted</i>)	Peratusan Kesilapan	Peratusan Ketepatan
Kurang Padat	2682	2671	0.41%	99.59%
Sederhana Padat	1715	1679	2.10%	97.90%
Padat	644	564	12.42%	87.58%

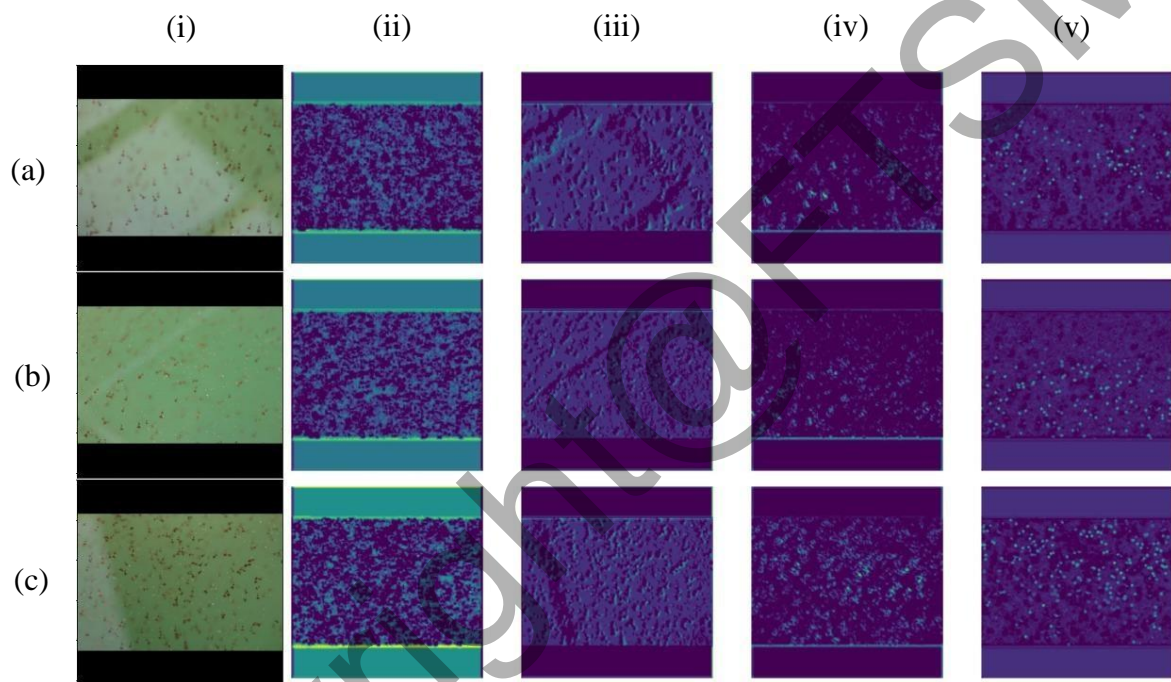
Jadual 7.5 : Keputusan peratusan ketepatan dan kesilapan berdasarkan jenis kepadatan

Jenis Kepadatan	Bilangan Sebenar (<i>Ground Truth</i>)	Bilangan Tekaan (<i>Predicted</i>)	Peratusan Ketepatan	Sisihan Piawaian
Kurang Padat	2682	2671	99.59%	0.02%
Sederhana Padat	1715	1679	97.90%	0.02%
Padat	644	564	87.58%	0.03%
Jumlah	5041	4914	97.48%	0.02%

Jadual 7.6 : Keputusan purata ketepatan dan sisihan piawaian berdasarkan jenis kepadatan

7.5 PETA KETUMPATAN (*DENSITY MAP*)

Idea utama untuk mengira objek secara tidak langsung dengan menganggar peta ketumpatan. Langkah pertama adalah menyediakan sampel latihan, supaya untuk setiap gambar terdapat peta kepadatan yang sesuai. Peta ketumpatan diperoleh dengan menerapkan konvolusi dengan kernel Gaussian (dan dinormalisasi sehingga mengintegrasikannya memberikan jumlah objek). Tujuannya adalah untuk melatih rangkaian konvolusional sepenuhnya untuk memetakan gambar ke peta kepadatan, yang kemudian dapat disatukan untuk mendapatkan jumlah objek. Peta ketumpatan mengikut kepadatan udang ditunjukkan seperti dalam Rajah 7.4.



Rajah 7.4 : (i) Imej dengan mengikut kepadatan udang dengan (a) kurang padat dengan bilangan sebenar 46 ekor dan bilangan tekaan 46 ekor (b) sederhana padat dengan bilangan sebenar 118 ekor dan bilangan tekaan 118 ekor (c) padat dengan bilangan sebenar 188 ekor dan bilangan tekaan 170 dan (ii) penapis untuk Resnet101 res2c_out, (iii) penapis untuk Resnet101 res3c_out, (iv) penapis untuk Resnet101 res4w_out dan (v) penapis untuk rpn_bbox

7.6 PERBANDINGAN KEPUTUSAN ANTARA BILANGAN SEBENAR DENGAN BILANGAN TEKAAN

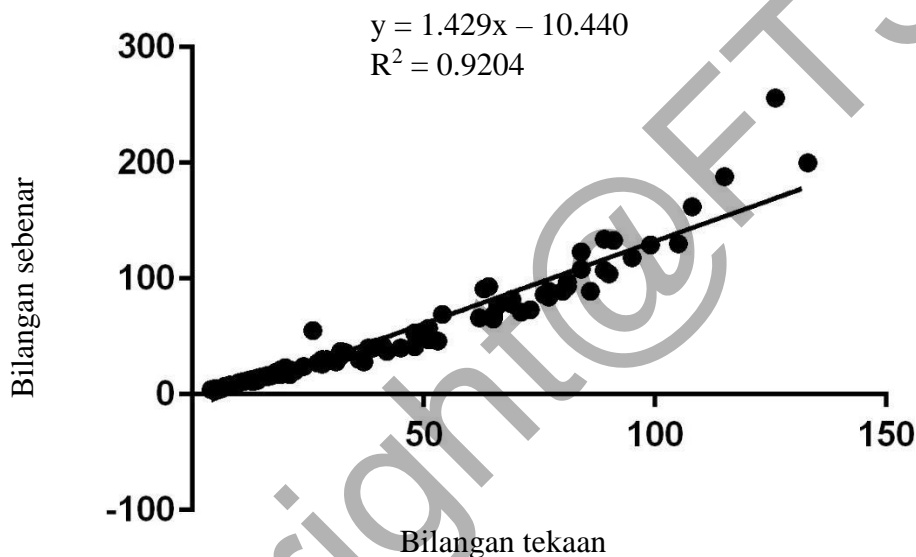
Kaedah yang dilakukan terhadap bilangan sebenar dengan bilangan tekaan adalah menggunakan regresi linear (*Linear Regression*). Dalam perbandingan antara bilangan sebenar dengan bilangan tekaan, nilai R^2 akan diuji. Dari nilai R^2 , akan mengetahui bahawa garis regresi sesuai dengan data yang digunakan atau tidak. Nilai R^2 yang berhampiran dengan nilai

sebanyak 1 merupakan garis regresi sesuai dengan data yang digunakan. Formula 7.8 menunjukkan formula untuk mengira nilai R^2 .

$$R^2 = \left(\frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{[\sum (x_i - \bar{x})^2] [\sum (y_i - \bar{y})^2]}} \right)^2$$

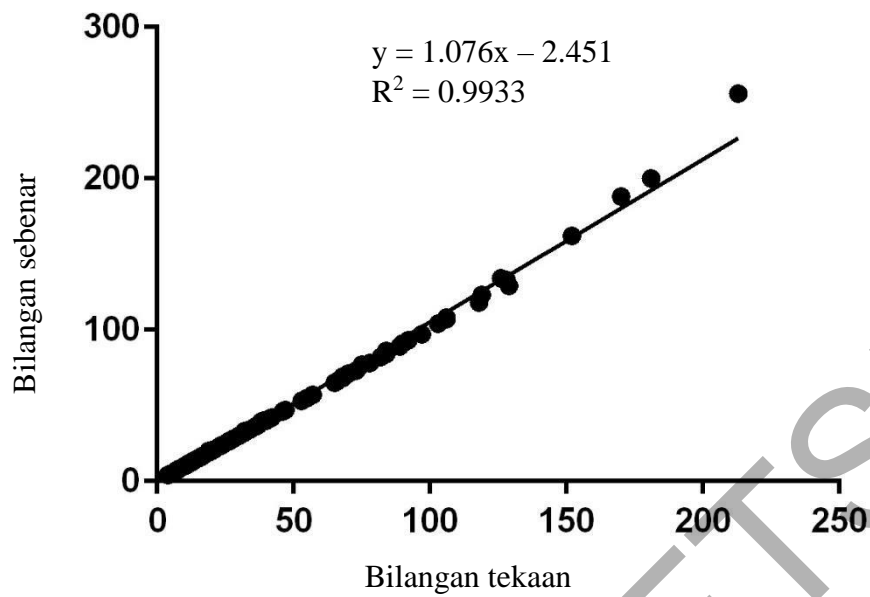
Formula 7.8 : Formula untuk mengira nilai R^2

Rajah 7.5 menunjukkan bahawa garis regresi tidak sesuai dengan data yang digunakan iaitu jumlah bilangan sebenar berbanding dengan bilangan tekaan.



Rajah 7.5 : Regresi linear antara bilangan tekaan dan bilangan sebenar untuk 100 imej bagi model asal

Rajah 7.6 menunjukkan bahawa garis regresi sesuai dengan data yang digunakan iaitu jumlah bilangan sebenar berbanding dengan bilangan tekaan. Nilai R^2 hanya mencatat sebanyak 0.9204 dalam model asal yang digunakan manakala model yang selepas ditambahbaik mencatat nilai sebanyak 0.9933.



Rajah 7.6 : Regresi linear antara bilangan tekaan dan bilangan sebenar untuk 100 imej bagi model yang selepas ditambahbaik

7.7 KESIMPULAN

Kesimpulannya, beberapa kaedah telah menguji terhadap model yang telah dihasilkan dan dapati keseluruhan prestasi model tersebut dapat memberi ketepatan yang amat baik. Dalam pemilihan *hyperparameter* yang optimum semasa fasa latihan model amat penting, supaya model yang dihasilkan dapat memberikan prestasi serta ketepatan yang baik.

BAB VIII

KESIMPULAN

8.1 PENGENALAN

Pembangunan Sistem Pengiraan Udang dapat menepati objektif projek yang mana telah dinyatakan. Pembangunan sistem ini diharapkan dapat membawa manfaat kepada penggunanya, khususnya petani udang dan pembekal udang. Dalam bab ini akan membincangkan kesimpulan bagi keseluruhan sistem yang akan dibangunkan. Kesimpulan yang akan dibincangkan merangkumi gambaran keseluruhan hasil secara menyeluruh Sistem Pengiraan Udang, kegunaan sistem dan cadangan penambahbaikan sistem di masa hadapan.

8.2 KEKANGAN

Berdasarkan kajian yang dijadikan terdapat beberapa kekangan yang dikenal pasti. Antaranya adalah :

- i) Data yang akan digunakan sebagai imej latihan perlu ditangkap disebabkan tidak dapat imej yang sedia ada dalam laman web.
- ii) Sistem ini perlu menggunakan komputer untuk membuka antara muka supaya proses pengiraan dapat dijalankan.
- iii) Sistem ini perlu menggunakan satu kamera yang mempunyai resolusi yang tinggi supaya sistem ini dapat mengesan udang lebih jelas dan mudah di dalam bekas.

8.3 PENAMBAHBAIKAN DI MASA HADAPAN

Penambahbaikan yang boleh dilakukan adalah pengguna boleh merekodkan data dengan menggunakan sistem ini seperti suhu air, dan data lain yang penting seperti nama pembeli serta jumlah bilangan pengeluaran udang selepas proses pengiraan telah diselesaikan. Di samping itu, penambahbaikan yang lain adalah sistem dapat mengeluarkan bunyi amaran semasa jumlah bilangan amaun udang yang dikira oleh sistem telah melebihi jumlah bilangan amaun udang yang ditetapkan oleh bekas. Akhir sekali, penambahbaikan yang dapat dibuat adalah

menjadikan sistem ini dengan atas talian serta data yang disimpan melalui SQL atau *Cloud Based* supaya urusan perniagaan dapat dimudahkan.

8.4 KESIMPULAN

Kesimpulannya, Sistem Pengiraan Udang yang akan dibangunkan dapat mencapai objektif kajian ini bagi menyelesaikan masalah utama yang berlaku semasa mengadakan proses pengiraan. Masalah utama yang berlaku adalah masa untuk proses pengiraan memerlukan masa yang panjang dan kebanyakan jumlah bilangan amaun udang dikira dengan cara manual didapati tidak tepat. Sistem yang akan dibangunkan juga diharap dapat memenuhi tahap keperluan pengguna iaitu petani udang dan pembekal udang.

RUJUKAN

- Ho Mei Kei. "Selected Agricultural Indicators, Malaysia 2015." Department of Statistics Malayisa, 2015.
- Ho Mei Kei. "Selected Agricultural Indicators, Malaysia 2016." Department of Statistics Malayisa, 2016.
- Ho Mei Kei. "Selected Agricultural Indicators, Malaysia 2017." Department of Statistics Malayisa, 2017.
- Kaiming He, Georgia Gkioxari, Piotr Dollar, Ross Girshick. "Mask R-CNN" Facebook AI Research (FAIR), 24 Jan 2018.
- Jonathan Long, Evan Shelhamer, Trevor Darrell, "Fully Convolutional Networks for Semantic Segmentation" UC Berkeley, 8 Mar 2015.
- Jonathan Gana Kolo, James Agajo, A. M. Aibinu, Abdullahi Muhammad Orire. "A Proposed Fish Counting Algorithm Using Digital Image Processing Technique" Federal University of Technology Minna, March 2017.
- Rollyn Labuguen, Albert Causo, Edsel Jose P. Volante, Roy Bayot. "Automated Fish Fry Counting And Schooling Behavior Analysis Using Computer Vision" March 2012.
- Francisco Silverio. "Automatic Fish Counting in Aquariums" University of Lisbon.
- Daniel Onoro-Rubio, Roberto J. Lopez-Sastre. "Towards perspective-free object counting with deep learning." GRAM, University of Alcala, Spain, 2016.

Jirabhorn Kaewchote, Sittichoke Janyong, Wasit Limprasert. "Image recognition method using Local Binary Pattern and the Random forest classifier to count post larvae shrimp." Agriculture and Natural Resource 52 371-376, 2018.

Victor Lempitsky and Andrew Zisserman. "Learning To Count Objects in Images." Visual Geometry Group University of Oxford, 2010.

Tobias Stahl, Silvia L. Pinteá and Jan C. van Gemert. "Divide and Count: Generic Object Counting by Image Divisions." 2017.

M Solahudin, Slamet W, Dwi A S. "Vaname (*Litopenaeus vannamei*) Shrimp Fry Counting Based on Image Processing Method." Department of Mechanical and Biosystem Engineering, Bogor Agricultural University, Indonesia, 2018.

Maryam Rahnemoonfar and Clay Sheppard. "Deep Count: Fruit Counting Based on Deep Simulated Learning." University-Corpus Christi, Corpus Christi, USA, 2017.

Mehmet Baygin, Mehmet Karakose, Alisan Sarimaden and Erhan Akin. "An Image Processing based Object Counting Approach for Machine Vision Application." Computer Engineering Department, Ardahan University, Ardahan & Firat University, Elazig, 2018.

Jonathan Hui. Image segmentation with Mask R-CNN.

https://medium.com/@jonathan_hui/image-segmentation-with-mask-r-cnn-ebe6d793272.

Dhruv Parthasarathy. A Brief History of CNNs in Image Segmentation : From R-CNN to Mask R-CNN. <https://blog.athelas.com/a-brief-history-of-cnns-in-image-segmentation- from-r-cnn-to-mask-r-cnn-34ea83205de4>.

Pulkit Sharma. Computer Vision Tutorial : Implementing Mask R-CNN for Image Segmentation. <https://www.analyticsvidhya.com/blog/2019/07/computer-vision-implementing-mask-r-cnn-image-segmentation/>.

Aditi Mittal. Instance segmentation using Mask R-CNN.

<https://towardsdatascience.com/instance-segmentation-using-mask-r-cnn-7f77bdd46abd>.

Gabriel Garza. Mask R-CNN for Ship Detection & Segmentation.

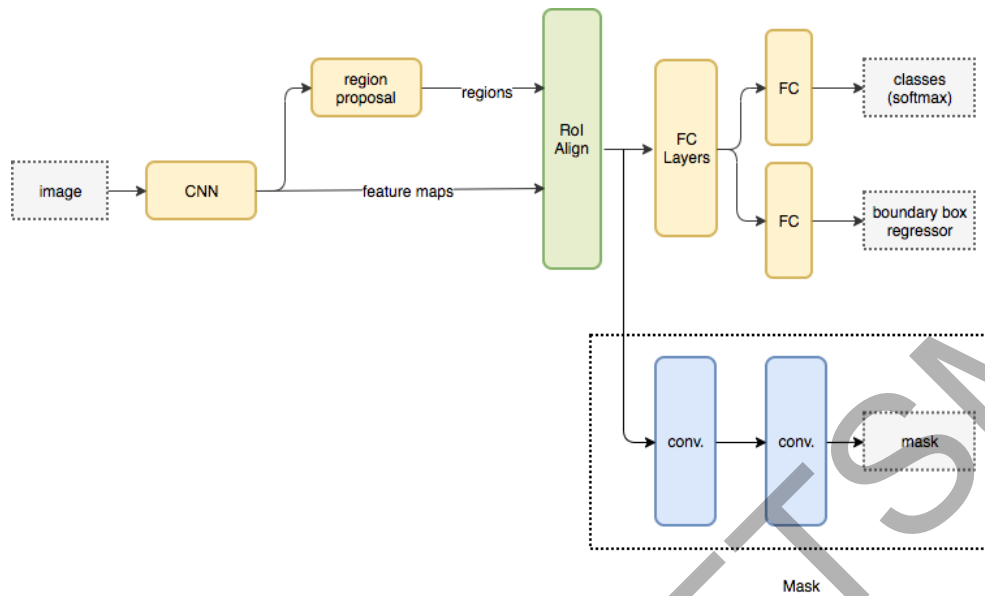
<https://towardsdatascience.com/mask-r-cnn-for-ship-detection-segmentation-a1108b5a083>.

Avadoot R. Telepatil, Prashant M. Jadhav, “Colour Object Counting and Sorting Mechanism Using Image Processing Approach”, Internasional Journal of Modern Trends in Engineering and Research.

Amit R. Chavan, Rajveer K. Shastri, Deosarkar Shankar, “Implementation of Object Counting using Image Processing Techniques in Animal Husbandry Field: A Review”, March 2013.

Mengxiao Tian, Hao Guo, Hong Chen, Qing Wang, Chengjiang Long, Yuhan Ma, “Automated pig counting using deep learning”, Computer and Electronics in Agriculture 163 (2019) 104840.

LAMPIRAN



LAMPIRAN A Adaptasi daripada (Medium : Jonathan Hui)

Merujuk ke Lampiran A, dalam Topeng R-CNN (Mask R-CNN) akan membahagikan kepada dua bahagian, iaitu segmentasi dan pengesanan.

Pseudocode Mask R-CNN :

Gambar yang dimasukkan

Memasukkan gambar dalam algoritma CNN

Membahagikan dua bahagian

1) Mendapatkan *region proposal* dan *regions*

2) Mendapatkan *feature maps*

Gabungkan keputusan daripada dua bahagian untuk mendapatkan *ROI Align*

Keputusan akan membahagikan dua bahagian lagi

1) Bahagian *FC Layers*

Membahagikan FC yang kecil untuk mendapatkan *classes* dan *boundary box regressor*

2) Bahagian *Mask*

Melalui 2 kali *layers Conv* dan mendapatkan *mask*