BAB II
TINJAUAN PUSTAKA

2.1 Penelitian Sebelumnya

Penelitian terkait yang dilakukan mengenai deteksi objek uang kertas menggunakan algoritma *Faster R-CNN* sudah banyak dilakukan sebelumnya, baik untuk mendeteksi objek uang dengan metode lain ataupun dengan menggunakan *Faster R-CNN* tetapi dengan objek yang lain. Tidak sedikit di antaranya memberikan hasil yang sesuai sehingga dapat diterapkan dalam masyarakat. Berikut merupakan penelitian terdahulu yang menurut penulis memiliki keterkaitan dengan penelitian yang akan dilakukan.

Penelitian pertama dengan judul "Implementasi Deep Learning Menggunakan Metode *Convolutional Neural Network* dan Algoritma YOLO dalam Sistem Pendeteksi Uang Kertas Rupiah Bagi Penyandang *Low Vision"* yang dilakukan oleh Kevin Maulana Azhar, Imam Santoso, dan Yosua Alvin Adi Soetrisno pada tahun 2021[21]. Penelitian ini menggunakan metode CNN yang menggunakan *framework* Darknet dengan algoritma YOLO dalam mendeteksi citra uang kertas rupiah. Dataset yang digunakan berupa citra uang kertas emisi 2016 yang terbagi menjadi 14 kelas dengan total data sebanyak 1400 gambar. Dataset dibagi menjadi data *train*, *validation*, dan *test* dengan bobot 70:20:10. Hasil dari penelitian ini menunjukkan model yang dibangun dengan metode CNN dan algoritma YOLO mendapatkan nilai mAP sebesar 88% dengan waktu respon untuk mendeteksi uang sebesar 1,28 detik.

Penelitian kedua berjudul "Pendeteksian Nominal Uang Pada Gambar Menggunakan *Convolutional Neural Network*: Integrasi Metode Pra-Pemrosesan Citra dan Klasifikasi Berbasis CNN" yang dilakukan oleh Muhamad Malik Ibrahim, Reni Rahmadewi, dan Lela Nurpulaela pada 2023 [22]. Penelitian ini menggunakan CNN dengan arsitektur MobileNetV2. Dataset yang digunakan berjumlah 1076 gambar yang terdiri dari 8 kelas. Hasil penelitian menunjukkan pengujian secara manual menghasilkan probabilitas secara berurutan pada kelas Rp1.000, Rp2.000, Rp5.000, Rp10.000, Rp20.000,

Rp50.000, Rp75.000, Rp100.000 sebesar 20%, 10%, 70%, 50%, 40%, 100%, 80%, 90%. Angka tersenut dihasilkan dari 10 kali pengujian di setiap kelasnya, kecuali kelas Rp75.000 sebanyak 5 kali.

Penelitian ketiga dengan judul "Implementasi Metode *Faster Region Convolutional Neural Network (Faster R-CNN)* Untuk Pengenalan Jenis Burung Lovebird" yang dilakukan oleh Fino Charli dkk. pada tahun 2020 [23]. Penelitian ini menggunakan dataset citra burung lovebird dengan 808 gambar dan 8 kelas dengan masing-masing 100 gambar di setiap kelasnya. Hasil dari penelitian ini yang menggunaan model *Faster R-CNN* menunjukkan hasil dengan tingkat akurasi model berada pada rentang 78% hingga 99%. Hal ini dapat disimpulkan bahwa metode *Faster* Region Convolutional Neural Network dapat memprediksi citra burung lovebird.

Penelitian keempat yang berjudul "Klasifikasi Pola Kain Tenun Melayu Menggunakan *Faster R-CNN*"[14] yang dilakukan oleh Yoze Rizki dkk. pada tahun 2021. Penelitian ini bertujuan untuk mengevaluasi kinerja klasifikasi motif tenun Melayu menggunakan metode *Faster R-CNN*. Pada penelitian ini, digunakan model arsitektur VGG dalam penerapan *Faster R-CNN*. Dataset yang digunakan terdiri dari 100 gambar yang telah diacak untuk setiap dari 5 lipatan dalam validasi silang *K-fold*. Data tersebut kemudian dibagi menjadi data pelatihan dan data pengujian dengan rasio 80:20. Hasil dari pelatihan model menunjukkan rata-rata nilai kehilangan pelatihan sebesar 1,915. Validasi menggunakan K-fold cross validation dengan nilai k=5 menghasilkan akurasi sebesar 82,14%, dengan presisi 91,38% dan recall 91,36%. Berdasarkan hasil ini, penggunaan *Faster R-CNN* dengan VGG terbukti unggul secara keseluruhan dibandingkan dengan algoritma serupa lainnya.

Penelitian kelima berjudul "Pendeteksian Objek Pada Citra Hewan Karnivora dan Herbivora Menggunakan *Faster R-CNN*" yang dilakukan oleh Sherien Trisnawaty Eka Putri dan Achmad Fahrurozi pada tahun 2022[24]. Dalam penelitian ini, sistem deteksi objek dikembangkan menggunakan metode *Faster R-CNN* untuk mengklasifikasikan jenis hewan karnivora dan herbivora berdasarkan citra. Pada tahap pelatihan, penelitian ini menggunakan

arsitektur *Inception* V2. Dataset yang digunakan berasal dari Google dan diberi label secara manual, dengan total 2000 gambar hewan. Model yang dihasilkan melalui proses *training* memiliki 50.000 langkah (steps). Hasil penelitian menunjukkan bahwa tingkat akurasi rata-rata untuk mendeteksi hewan karnivora dan herbivora adalah 89%.

Penelitian keenam berjudul "Identifikasi Sampah Plastik Menggunakan Algoritma *Deep Learning*" yang dilakukan oleh Lut Faizal, Yuyun, dan Hazriani pada tahun 2023[25]. Dalam penelitian ini, dataset yang digunakan diperoleh melalui pengambilan data dari situs web dan langsung dari tempat wisata di Kota Makassar. Total data yang diperoleh berjumlah 991 gambar. Data kemudian mengalami proses *preprocessing* dengan meresize gambar menjadi ukuran 540 x 540 piksel. Selanjutnya, dataset dibagi menjadi data uji dan data latih dengan perbandingan 80:20. Metode yang digunakan dalam penelitian ini adalah model *Faster R-CNN* dengan arsitektur ResNet-50 COCO. Nilai *learning rate* yang digunakan adalah 0,04, dengan step 100k, dan hasil dari proses *training* menunjukkan nilai *loss* sebesar 0,008. Waktu *training* yang diperlukan adalah 8 jam 12 menit. Hasil penelitian menunjukkan total akurasi sebesar 96,3% untuk mengenali 5 objek yang diuji.

Penelitian ketujuh berjudul "Deteksi *Non-Spoofing* Wajah pada Video secara *Real Time* Menggunakan *Faster R-CNN*" yang dilakukan oleh Sunario Megawan dkk., pada tahun 2022[26]. Penelitian ini bertujuan mengembangkan model *Faster R-CNN* yang mampu mendeteksi wajah *non-spoof* dan *spoof* secara *real time*. Model ini menggunakan arsitektur CNN ResNet-50 dengan learning rate sebesar 0,00001 dan 71 *epoch*. Data yang digunakan adalah data dalam bentuk video yang terdiri dari data *non-spoof* maupun *spoof* terdiri dari masing-masing 4 video dengan 193 frame di setiap videonya dan durasi 12 detik. Penelitian ini menghasilkan akurasi sebesar 92% untuk deteksi wajah *non-spoof* dan 96% untuk deteksi wajah *spoof* secara real time dengan *frame rate* 1 fps.

Tabel 2.1 Ringkasan Penelitian Sebelumnya

No.	Judul	Comparing	Constrating	Criticize	Synthesize	Summary
1	Implementasi Deep Learning Menggunakan Metode *Convolutional Neural Network* dan Algoritma YOLO dalam Sistem Pendeteksi Uang Kertas Rupiah Bagi Penyandang	Penelitian ini menggunakan metode CNN dengan *framework Darknet* dan juga algoritma YOLO. Data yang digunakan sebanyak 1400 gambar.	Penelitian ini bertujuan untuk melakukan deteksi uang kertas guna membantu penyandang *Low Vision* dengan menerapkan metode CNN dengan algoritma YOLO.	Dalam penelitian tidak dijelaskan penetapan *threshold* yang digunakan dalam perhitugan IoU.	Dataset yang digunakan bersumber dari foto pribadi dan *google images* yang terbagi menjadi 14 kelas yang terdiri dari 7 nominal uang kertas rupiah emisi 2016. Dataset yang dikumpulkan dibagi menjadi 3 data *train*, *valdation*, dan *testing* dengan bobot 70:10:10. *Training* data menggunakan framework *darknet* sebanyak 7000 iterasi.	Hasil dari penelitan ini model yang digunakan dapat melakukan detksi uang kertas rupiah dengan waktu respon sebesar 1,28 detik dan menghasil nilai mAP sebsar 88%.

No.	Judul	Comparing	Constrating	Criticize	Synthesize	Summary
	Low Vision[21]					
2	Pendeteksian Nominal Uang Pada Gambar Menggunakan *Convolutional Neural Network*[22]	Penelitian ini menggunakan metode CNN dengan MobileNetV2. Dataset yang digunakan sebanyak 1076.	Dalam penelitian ini, sistem untuk mendeteksi uang kertas dalam citra dirancang menggunakan arsitektur *deep learning* berbasis *Convolutional Neural Network* (CNN).	Dalam penelitian ini hanya digunakan pengujian secara manual tanpa menggunakan *evaluation metrics*, yang mana hasil pengujiannya tidak cukup menggambarkan model yang dibangun. Selain itu, dalam proses *training* tidak	Dataset yang digunakan sebanyak 1076 yang terdiri 8 kelas uang kertas rupiah dengan emisi 2016 dan 2022. Model yang dibangun menggunakan *activation function softmax*, *optimizer adam*, serta menggunakan *loss function categorical cross-entropy*.	Hasil penelitian menunjukkan pengujian secara manual menghasilkan probabilitas secara berurutan pada kelas Rp1.000, Rp2.000, Rp5.000, Rp10.000, Rp20.000, Rp50.000, Rp75.000, Rp100.000 sebesar 20%, 10%, 70%, 50%, 40%, 100%, 80%, 90%. Angka tersenut dihasilkan dari 10 kali pengujian di setiap kelasnya, kecuali kelas

No.	Judul	Comparing	Constrating	Criticize	Synthesize	Summary
				diberikan visualisasi terkait model yang digunakan. Dataset yang digunakan bisa dikatakan sangat *imbalance.*		Rp75.000 sebanyak 5 kali.
3	Implementasi Metode *Faster Region Convolutional Neural Network* (*Faster R-CNN*) Untuk Pengenalan	Penelitian ini menggunakan metode *Faster R-CNN* untuk melakukan pemodelan dengan jumlah data sebanyak 808.	Penelitian ini bertujuan untuk menerapkan metode *Faster R-CNN* untuk mengenali jenis burung *lovebird* untuk mengetahui motif asli dari jenis burung	Penulis tidak memberikan detail lebih lanjut mengenai proses *training*. Penulis hanya menuliskan total *step*. Penulis juga tidak memberikan hasil	Pada penelitian ini hanya diambil 8 jenis dari total 9 jenis burung *lovebird* yang ada. Masing-masing kelas berjumlah 100 gambar. Penulis membagi dataset menjadi 80% data *train* dan 20% data *test*. Format gambar adalah .jpg dilakukan agar data uji dan data latih	Tingkat akurasi model yang didapatkan dari hasil pendeteksian jenis burung *lovebird* pada suatu citra digital menggunakan *Faster Region Convolutional Neural Network* berkisar 78% hingga 99%

No.	Judul	Comparing	Constrating	Criticize	Synthesize	Summary
	Jenis Burung *Lovebird*[23]		*lovebird* berdasarkan tingkat akurasi deteksi *object*.	*training* setiap kelasnya."	mendapat bobot *training* yang sama sehingga menghasilkan bias rendah pada data latih dan data uji. *Training* dilakukan sebanyak 200000 step. *Training* dilakukan dengan *Faster R-CNN* hingga nilai loss konsisten di bawah 0,05.	
4	Klasifikasi Pola Kain Tenun Melayu Menggunakan *Faster R-CNN*[14]	Penelitian ini menggunakan metode *Faster R-CNN* untuk melakukan pemodelan dengan jumlah data 100 gambar kain tenun	Penelitian ini bertujuan untuk mengetahui performa dari *Faster R-CNN* dan arsitektur VGG dalam melakukan	Pada penelitian penulis sudah menjelaskan secara detail mengenai proses yang dilakukan.	Dalam penelitian ini, dataset terdiri dari dua kelas, yaitu "pucuk rebung" dan "siku keluang". Pada tahap *preprocessing*, dilakukan augmentasi data dengan memberikan distorsi pada sampel citra dan juga memperlebar citra. Dataset	Dalam penelitian ini, pengenalan karakteristik motif tenun Melayu diklasifikasikan menggunakan metode deteksi objek *Faster Region-based Convolutional Neural Network* (*Faster R-CNN*)

No.	Judul	Comparing	Constrating	Criticize	Synthesize	Summary
		melayu di setiap kelasnya.	klasifikasi motif tenun melayu.		kemudian dibagi menjadi delapan kelas dan terbagi menjadi data latih (*train*) dan data uji (*test*) dengan perbandingan 80:20. Hasil rata-rata dari *training loss* adalah 1,915.	dengan arsitektur VGG. Hasil validasi K-Fold *Cross Validation* dengan nilai k=5 menunjukkan akurasi sebesar 82,14%, presisi 91,38%, dan *recall* 91,36%.
5	Pendeteksian Objek Pada Citra Hewan Karnivora dan Herbivora Menggunakan *Faster R-CNN* dengan	Penelitian ini menggunakan metode *Faster R-CNN*, menggunakan arsitektur *Inception V2*.	Peneitian ini bertujuan untuk menghasilkan sistem deteksi objek untuk kasifikasi hewan karnivora dan herbivora menggunakam *Faster R-CNN*	Akurasi rata-rata sebesar 89%. Namun, pada kelas *camel* dan zebra menghsilkan akurasi yang lebih rendah yaitu sebesar 70%.	Pada penelitian ini data yang digunakan merupakan data hewan karnivora dan herbivora yang diperoleh melalui *google chrome* sebanyak 2000 citra hewan. Dataset dalam penelitian ini terbagi menjadi 20 kelas dengan 10 kelas hewan karnivora dan 10 hewan herbivora. *Batch size* yang	Hasil penelitian menunjukkan bahwa sistem pendeteksian gambar hewan karnivora dan herbivora rata-rata memiliki akurasi sebesar 89%. Berdasarkan hasil *Recall* dan *Precision*, kinerja sistem termasuk dalam kategori baik, dengan nilai *Recall* 100%

No.	Judul	Comparing	Constrating	Criticize	Synthesize	Summary
	arsitektur *Inception V2*[24]		dengan arsitektur *Inception V2*		digunakan adalah 1 dan data dalam penelitian ini dilatih sebanyak 50.000 *steps*	pada hewan Cheetah, Eagle, Komodo, Shark, *Tiger*, *Bull*, Guineapig, dan Zebra.
6	Identifikasi Sampah Plastik Menggunakan Algoritma *Deep Learning*[25]	Penelitian ini menggunakan metode *Faster R-CNN* dengan model ResNet-50 COCO, nilai *learning rate* 0,04 dan *step* sebanyak 100k.	Penelitian ini bertujuan untuk melakukan identifikasi sampah jenis plastik menggunakan *Faster R-CNN*.	Penulis tidak memberikan gambaran umum mengenai contoh citra pada dataset yang digunakan. Waktu *train* yang dilakukan juga cukup panjang selama 8 jam 12 menit.	Dataset yang digunakan dalam penelitian ini diperoleh melalui pengambilan data dari situs web dan pengambilan gambar di Kota Makassar. Dataset ini dibagi menjadi dua bagian: data latih (*train*) dan data uji (test) dengan perbandingan 80:20. Selain itu, dataset ini terdiri dari lima kelas sampah dengan ukuran gambar 540x540 piksel.	Dari hasil penelitian, sistem memiliki tingkat akurasi sebesar 96,3% dalam mengidentifikasi objek. Selain itu, sistem juga dapat memperkirakan berat dari objek yang terdeteksi berdasarkan parameter *input* dari basis data.
7	Deteksi *Non-*	Penelitian ini menggunakan	Penelitian ini bertujuan untuk	Dalam penelitian ini tidak	Data pelatihan yang digunakan dalam penelitian	Pada tahap pengujian, diperoleh akurasi sebesar

No.	Judul	Comparing	Constrating	Criticize	Synthesize	Summary
	Spoofing Wajah pada Video secara *Real Time* Menggunakan *Faster R-CNN*[26]	*Faster R-CNN* dengan arsitektur CNN ResNet-50 dengan *learnung rate 0,00001* dan *epoch* 71.	membuat model *Faster R-CNN* yang dapat mendeteksi *non-spoof* dan *spoof* wajah secara *real time* menggunakan *Raspberry Pi.*	dijelaskan lebih detail mengenai arsitektur ResNet-50 yang digunakan. Dalam penelitian ini hanya disebutkan satu kali tanpa penjelasan lebih lanjut.	ini merupakan data primer yang diperoleh menggunakan kamera dalam format video. Data pelatihan disajikan dalam bentuk video yang menampilkan satu wajah menghadap ke depan dengan latar belakang gambar tanpa foto manusia. Baik data non-spoof maupun spoof terdiri dari masing-masing 4 video dengan 193 *frame* per video dan durasi 12 detik.	92% untuk deteksi wajah *non-spoof* dan 96% untuk deteksi wajah *spoof* secara *real-time* dengan kecepatan *frame* 1 fps.

Berdasarkan penelitian terdahulu yang telah dijabarkan di atas, dapat disimpulkan bahwa penggunaan metode *Faster R-CNN* dalam melakukan deteksi objek dapat digunakan dan mendapatkan hasil yang baik. Pada penelitan 1 dan 2 digunakan sebagai referensi penelitian yang menggunakan objek berupa uang kertas rupiah dengan metode CNN. Pada penelitian 3 sampai 5 digunakan sebagai penerapan metode *Faster R-CNN* dalam objek penelitian yang bukan uang kertas rupiah, sedangkan pada penelitian 6 dan 7 dijadikan acuan sebagai penerapan *Faster R-CNN* dengan arsitektur ResNet-50.

2.2 Landasan Teori

Landasan teori mengandung informasi yang relevan dengan penelitian yang sedang dilakukan. Tujuannya adalah membantu peneliti dalam menjalankan penelitian tersebut. Berikut adalah landasan teori yang terkait dengan penelitian ini.

2.2.1 *Artificial Intelligence*

Artificial intelligence atau kecerdasan buatan adalah bidang dalam ilmu komputer yang mempelajari bagaimana mengotomatisasi perilaku cerdas. Teknologi ini memungkinkan komputer untuk berpikir dan meniru proses pembelajaran manusia, sehingga dapat menyerap informasi baru dan menerapkannya sebagai panduan di masa depan[27]. Secara historis, terdapat 4 (empat) pendekatan AI, yaitu: (1) *Thinking Humanly, Thinking Rationally, Acting Humanly, dan Acting Rationally.*

1. *Thinking Humanly*

   Thinking humanly merupakan aktivitas yang diakitkan dengan pemikiran manusia, seperti dalam pengambilan keputusan, pemecahan masalah, dan juga pembelajaran.

2. *Thinking Rationally*

   Thinking rationally merupakan studi mengenai komputasi yang memungkinkan untk memahami, berlogika, serta bertindak.

3. *Acting Humanly*

Acting humanly merupakan studi mengenai bagaimana cara komputer dapat melakukan hal-hal yang saat ini dilakukan oleh manusia dengan lebih baik.

4. *Acting Rationally*

   Acting Rationally merupakan kecerdasan mengenai desain *intelligent agent.*

2.2.2 *Machine Learning*

Machine learning merupakan cabang ilmu dari *artificial intelligence*. *Machine learning* memiliki kemampuan untuk membuat komputer berperilaku seperti manusia dan dapat meningkatkan pemahamannya melalui pengalaman secara otomatis[28]. *Machine learning* juga dapat diartikan sebagai sebuah program yang dapat belajar secara mandiri tanpa diprogram secara eksplisit[29]. Berdasarkan cara belajarnya, *machine learning* dibagi menajdi tiga cabang:

1. *Supervised Learning*

   Supervised learnig merupakan teknik dalam machine learning di mana mesin akan mempelajari data yang sudah diberikan label.

2. *Unsupervised Learning*

   Tidak seperti *supervised learning*, *unsupervised learning* adalah pendekatan di mana pembelajaran dilakukan tanpa panduan atau label. Dalam *unsupervised learning*, kita memiliki sejumlah *input* tanpa variabel output yang terkait.

3. *Reinforcement Learning*

   Reinforcement learning adalah metode di mana komputer berinteraksi dengan lingkungan yang dinamis dan harus mengeksekusi tugas tertentu[29].

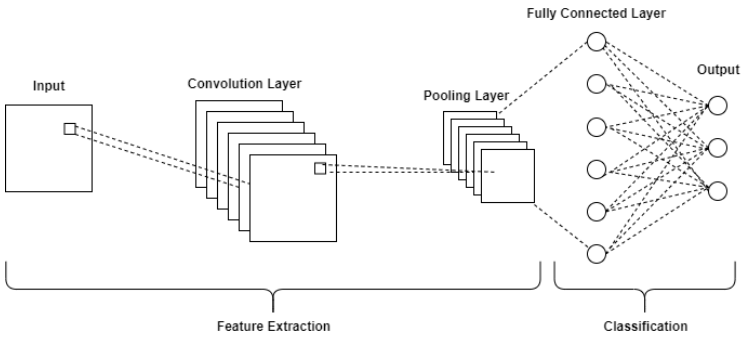
2.2.3 *Computer Vision*

Computer Vision atau yang dalam bahasa Indonesia disebut visi komputer merupakan cabang dari *Artificial Intelligence* dapat melatih komputer untuk bisa menerima informasi berupa gambar, video, ataupun *input* lain yang berhubungan dengan informasi visual. Sistem kerja visi komputer yaitu dengan

memproses data gambar yang diinputkan dan menggunakan kombinasi algoritma dalam pengolahan citra (*Image Processing)* serta kecerdasan buatan sehingga dapat menghasilkan suatu informasi dari citra tersebut[30].

2.2.4 *Convolutional Neural Network*

Convolutional Neural Network merupakan matriks yang memiliki sebuah fungsi untuk melakukan *filtering* terhadap sebuah gambar[31]. CNN termasuk ke dalam *Neural Network* yang merupakan salah satu arsitekur dalam *deep learning* yang terinspirasi dari otak manusia. CNN memiliki banyak sel untuk melakukan komputasi yang disebut dengan '*neuron*' [32]. Berikut merupakan arsitektur umum dari CNN:



Gambar 2.1 Arsitektur umum CNN

Secara umum, arsitrektur CNN terdiri atas *input layer*, *convolution layer*, *pooling layer*, dan juga *fully connected layer*.

2.2.4.1. *Input Layer*

Input layer merpukan lapisan dalam CNN yang mewakili data citra yang dimasukkan, data dalam *input layer* akan diubah ke dalam bentuk *array*[33]. Misalkan pada suatu gambar memiliki *input* dengan ukuran 64 x 64 px yang berjenis RGB, maka gambar tersebut akan memilki *input* berupa *array* dengan ukuran 64 x 64 x 3, di mana angka 3 mewakili jumlah *channel* dari gambar yaitu *red, green, blue*.

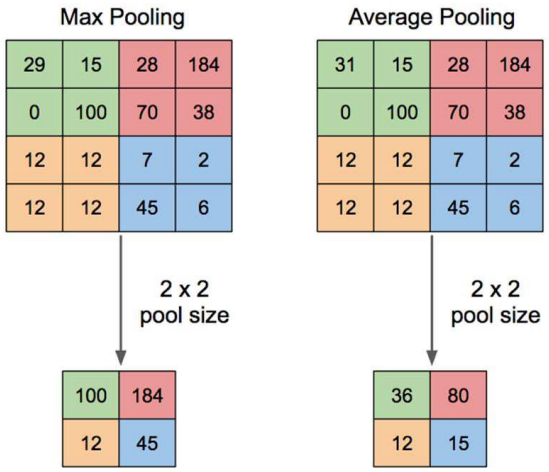
2.2.4.2. *Convolutional Layer*

Convolutional layer merupakan *layer* utama dari CNN. Tujuan dari *convolutional layer* adalah untuk melakukan ekstraksi *filter*. Gambar *input*

yang semula memiliki ukuran besar akan dibagi menjadi bagian-bagian gambar yang lebih kecil [34].

2.2.4.3. *Pooling Layer*

Pooling Layer merupakan *layer* yang berfungsi untuk melakukan pengurangan ukuran matriks[35]. *Layer* ini menggunakan operasi *down-sampling* sehingga data lebih mudah diatur pada proses selanjutnya. Salah satu jenis *pooling* yang paling sering digunakan adalah *max pooling* dan *average pooling*. *Maxpooling* mengambil nilai maximum dari setiap *region*, sedangkan *average pooling* mengambil nilai rata-rata dari setiap *region* Berikut adalah ilustrasi dari *maxpooling.*



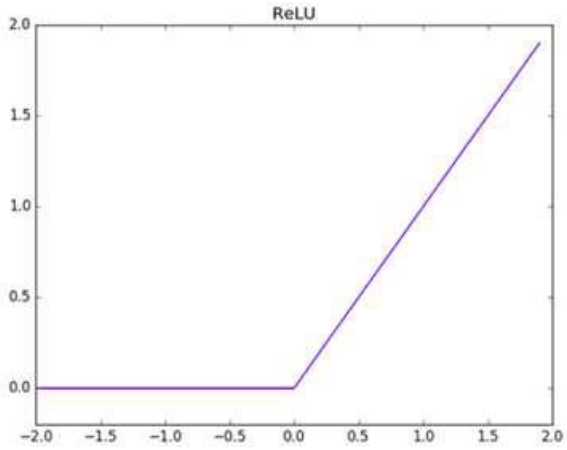
Gambar 2.2 Ilustrasi *max pooling* dan *average pooling* [36]

2.2.4.4. ReLU *activation*

Rectified Linear Unit (ReLU) merupakan jenis fungsi aktivasi dengan perhitungan yang sederhana. Saat melewati ReLU baik saat proses maju (*forward*) maupun saat proses mundur (*backward*), hanya digunakan kondisi if sederhana. ReLu memiliki persamaan sebagai berikut:

$$f(x) = \max(0, x) \tag{2.1}$$

ReLU didefinisikan sebagai fungsi aktivasi yang memetakan nilai x jika x > 0, selain itu apabila nilai x negatif diubah menjadi 0 [37].



Gambar 2.3 Fungsi aktivasi ReLU[36]

2.2.4.5. *Fully Connected Layer*

Fully Connected Layer merupakan lapisan di mana setiap *neuron* aktivasi dari lapisan sebelumnya terhubung dengan *neuron* pada lapisan berikutnya. Biasanya, *Fully Connected Layer* ditempatkan setelah proses konvolusi dan penggabungan (*pooling*) lapisan. Hasil dari proses ini berupa peta fitur (*feature map*), yang kemudian menjadi masukan bagi *Fully Connected Layer*[38].

2.2.4.6. Aktivasi *Softmax*

Softmax merupakan fungsi aktivasi yang digunakan untuk melakukan klasifikasi yang lebih dari dua kelas. *Softmax* biasanya digunakan dalam lapisan akhir untuk mempredikisi kelas dari gambar *input*. Dalam perhitungan probabilitas, fungsi aktivasi *softmax* menentukan klasifikasi multi kelas dengan output kelas dengan nilai probabilitas tertinggi. *Output* fungsi aktivasi *softmax* memiliki nilai probabilitas antara 0 dan 1 [39]. *Softmax* memiliki persamaan sebagai berikut:

$$softmax(z_i) = \frac{\exp(z_i)}{\sum_{j=1}^{n} exp(z_j)} \quad (2.2)$$

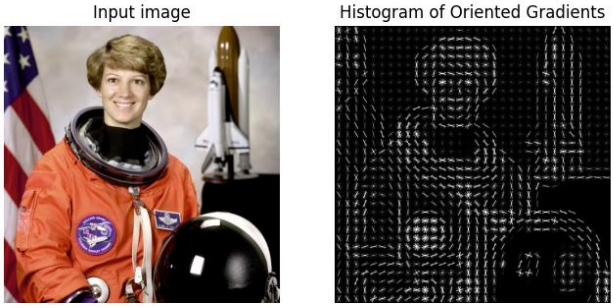
Z mewakili nilai dari neuron pada lapisan *output*. *Exp* merupakan eksponensial yang berperann sebagai fungsi *non-liniar*. Nilai tersebut dibagi dengan jumlah nilai eksponensial untuk dinormalisasi dan diubah menjadi probabilitas antara 0 dan 1.

2.2.5 *Hue Saturation Value (HSV)*

Hue Saturation Value (HSV) adalah format gambar yang menggambarkan warna berdasarkan kombinasi lingkaran warna. *Hue* menentukan sudut warna, di mana merah berada pada 0 derajat, hijau pada 120 derajat, dan biru pada 240 derajat. *Saturation* menunjukkan intensitas warna, dari gelap di tengah hingga putih di pinggir. Sementara itu, *Value* mengukur tingkat kecerahan warna, berkisar dari 0 hingga 100%[19].

2.2.6 *Histogram of Oriented Gradient (HOG)*

Histogram of Oriented Gradients (HOG) adalah sebuah teknik deskriptor fitur yang digunakan dalam visi komputer dan pemrosesan gambar untuk mendeteksi objek. Teknik ini berfokus pada bentuk objek dengan menghitung frekuensi orientasi gradien di setiap area lokal. Selanjutnya, HOG menghasilkan histogram berdasarkan magnitudo dan orientasi gradien tersebut[40].

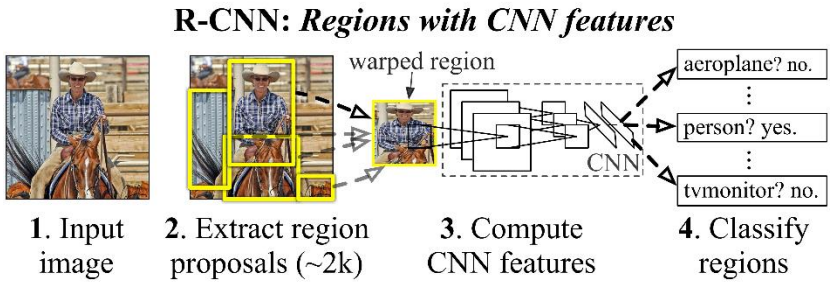


Gambar 2.4 Visualisasi HOG[41]

2.2.7 R-CNN

Region-based Convolutional Neural Network (*R-CNN)* merupakan algoritma yang diusulkan oleh Girshick dkk. pada tahun 2014. Hal ini didasarkan pada algoritma *selective search* untuk mengekstrak hanya 2000 wilayah yang dikenal sebagai *region proposal* dari sebuah gambar. Pendekatan ini membantu menghindari pengklasifikasian wilayah dalam jumlah besar. Masalah utama R-CNN terkait dengan kelemahan seperti pelatihan bersifat multi-tahap sehingga proses *training* membutuhkan

komputasi yang tinggi dan memakan waktu yang lama. Selain itu untuk dalam hal deteksi objek sangat lambat[9].

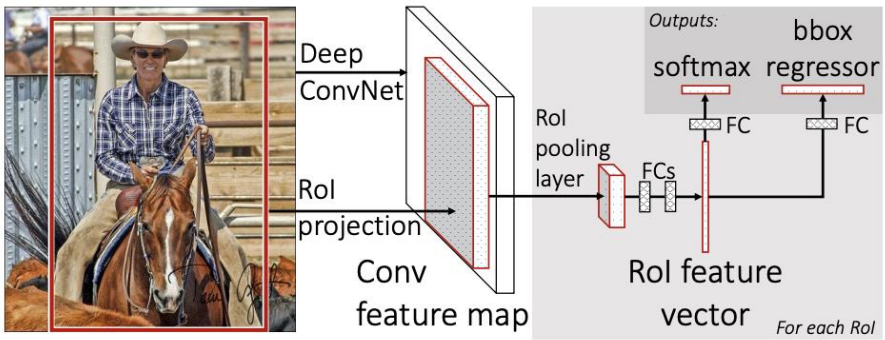


Gambar 2.5 Struktur dari R-CNN[42]

Proses R-CNN dimulai dengan melakukan *input* gambar yang dilakukan ekstraksi *region proposal* melalui proses *selective search.* Proses ini merupakan tahapan dasar dari R-CNN yang tujuannya adalah untuk menghasilkan proposal wilayah yang kemungkinan mengandung objek. Wilayah-wilayah tersebut dilakukan pemetaan ulang sebelum diproses dalam CNN. Fitur yang dihasilkan dari CNN dilakukan klasifikasi untuk menentukan kelasnya.

2.2.8 *Fast R-CNN*

Fast R-CNN adalah peningkatan dari model *R-CNN*. *Fast R-CNN* dapat melakukan komputasi 25 kali lebih cepat dibandingkan dengan *R-CNN*. Kecepatan ini disebabkan oleh perbedaan lokasi *region of interest*[12]. Pada Fast R-CNN, tidak perlu membuat hingga 2000 proposal dari sebuah *input*, karena jaringan saraf hanya perlu dijalankan sekali pada seluruh gambar[13].

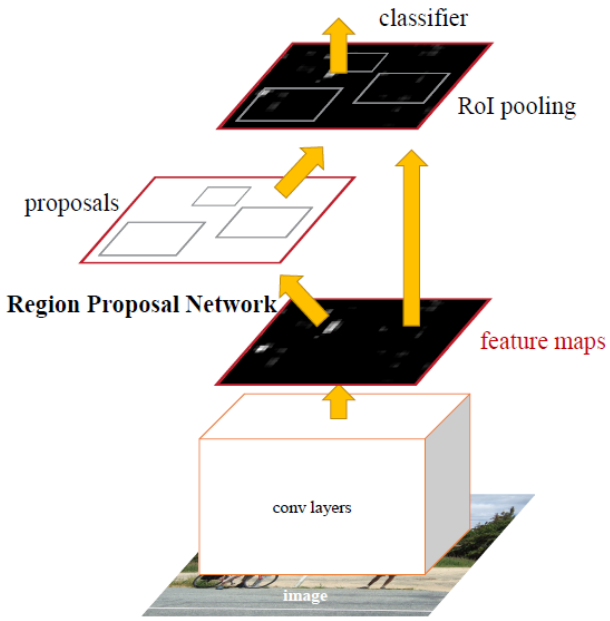


Gambar 2.6 Struktur dari *Fast R-CNN*

Proses *Fast R-CNN* diawali dengan *input layer* berupa citra yang kemudian diproses melalui CNN untuk menghasilkan *feature map.* Pada *Region of Interest* (RoI) *projection* diidentifikasi pada *input,* kemudian daerah-daerah tersebut diproyeksikan pada *feature map* konvolusi. Lapisan RoI *pooling* berfungsi untuk melakukan ekstraksi feature dengan ukuran tetap untuk setiap RoI dari *featrue map* yang dihasilkan dari proses konvolusi. *Feature map* dengan ukuran tetap kemudian diratakan dan diproses dalam *fully connected* untuk menghasilkan vektor dari fitur RoI. Pada lapisan *output,* vektor RoI *feature* diteruskan ke dua lapisan output, yaitu lapisan *softmax* untuk melakukan klasifikasi serta *box regressor* untuk memperbaiki koordinat *bounding box* dari setiap RoI.

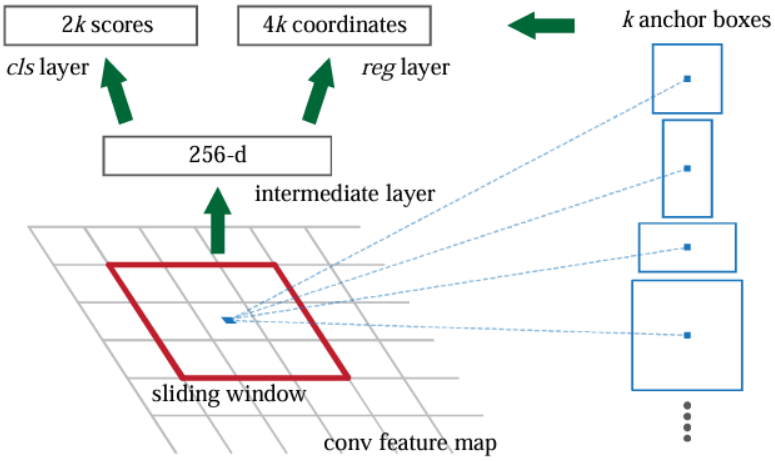
2.2.9 *Faster R-CNN*

Faster Region Based Convolutional Neural Network atau biasa disebut *Faster R-CNN* adalah algoritma dalam *object detection* yang merupakan penyempurnaan dari metode sebelumnya yaitu *R-CNN* dan juga *Fast R-CNN* yang pada metode tersebut digunakan Selective Search sebagai bentuk *input* yang akan diproses dalam CNN. Penggunaan selective search ditemukan kendala yang dapat menghambat proses komputasi yang memakan waktu lebih lama[43]. Pada metode *Faster R-CNN* peran *selective search* diganti dengan memanfaatkan *Region Proposal Network* (RPN). Berikut merupakan arsitektur dari *Faster R-CNN*.



Gambar 2.7 Arsitektur *Faster R-CNN* [44]

Region Proposal Network merupakan elemen kunci yang ada dalam jaringan *Faster R-CNN*. Fungsi utama dari RPN adalah untuk melakukan prediksi proposal daerah yang berpotensi mengandung objek dalam gambar *input*.

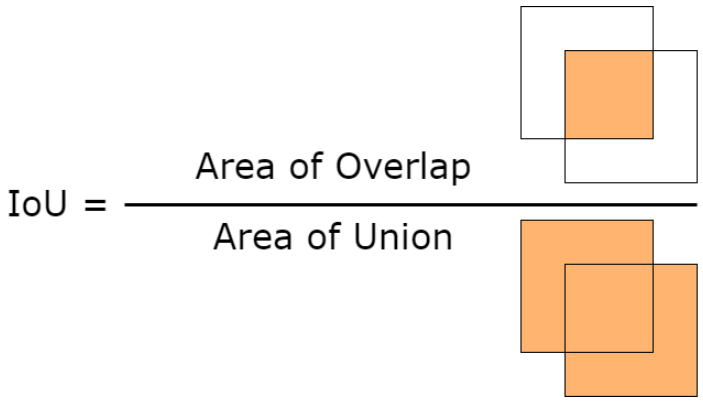


Gambar 2.8 Struktur dari RPN[44]

Pada RPN, terdapat sebuah *sliding window* dengan ukuran tetap yang diterapkan pada *feature map* yang dihasilkan oleh *convolutional layers*. *Sliding window* tersebut bekerja dengan melintasi *feature map* untuk

menghasilkan *region proposal.* Setiap *sliding window* menghasilkan vektor dengan *feature* 256 melalui lapisan *intermediate*. Pada setiap *sliding window* terdapat sejumlah *anchor boxes* dengan berbagai skala yang berfungsi untuk mengusulkan kandidat *bounding boxes.* Hasil dari *intermediate layer* berupa *classification layer* dan *regression layer. Classification layer* berfungsi untuk menunujukkan probabilitas apakah setiap kotak terdapat objek atau tidak. *Regression layer* berfungsi untuk memperbaiki koordinat *bounding box* yang diusulkan oleh *anchor boxes.* Hasil dari RPN adalah daftar proposal daerah yang selanjutnya digunakan oleh komponen deteksi objek yang ada pada *Faster R-CNN* untuk mengklasifikasikan dan mengidentifikasi objek dalam gambar. Proposal-proposal yang dihasilkan oleh RPN kemudian dibandingkan dengan *Ground Truth Box* menggunakan *Intersection Over Union*.

IoU (*Intersection over Union*) adalah istilah yang digunakan untuk menggambarkan sejauh mana tumpang tindih dua kotak. Semakin besar wilayah tumpang tindihnya, semakin besar pula nilai IoU-nya[45]. Perhitungan IoU adalah seperti dalan gambar berikut.

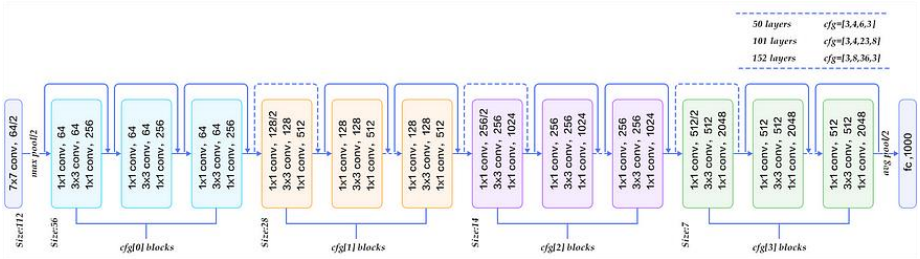


Gambar 2.9 Perhitungan IoU

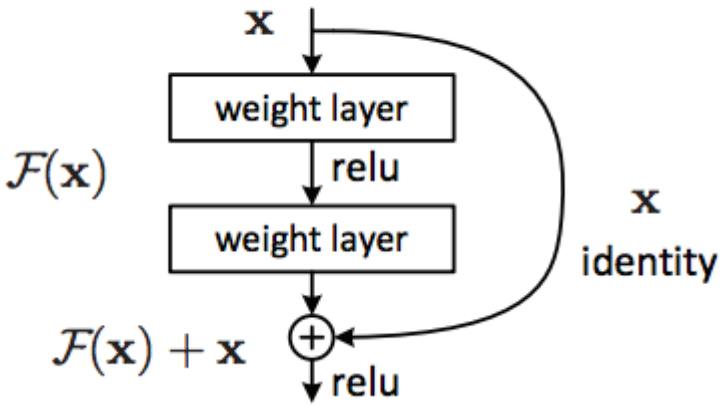
IoU dihitung dengan membagi area yang tuimpang tindih (*intersection*/*overlap*) dengan area gabungan dari seluruh area yang dicakup baik prediksi maupun *ground truth* (*area of union*).

2.2.10 *ResNet* (*Residual Network*)

ResNet merupakan arsitektur yang dibangun oleh Kaiming He dkk., yang menjadi juara satu kompetisi ILSVRC 2015. ResNet merupakan salah satu arsitektur CNN yang dikenal memiliki akurasi yang baik, karena menggunakan metode yang berbeda dari arsitektur CNN sebelumnya, yaitu *residual blocks dan skip connection*[46]. Salah satu varian dari arsitektur ResNet adalah ResNet-50. Dalam ResNet, diterapkan konsep koneksi lompat (skip connection), di mana fitur yang merupakan *input* dari lapisan sebelumnya juga digunakan sebagai *input* untuk lapisan berikutnya[47]. Berikut meupakan arsitektur umum ResNet-50:



Gambar 2.10 Arsitektur ResNet-50 [48]



Gambar 2. 11 *Residual Block* pada ResNet-50 [48]

2.2.11 *Negative Log-Likelihood (NLL)*

Negative Log-likelihood merupakan metrik loss yang digunakan untuk mengukur seberapa baik model memprediksi distribusi data. Metrik ini dihitung dengan menjumlahkan *log-likelihood* probabilitas data yang diamati,

di mana probabilitas dihitung berdasarkan prediksi model. Semakin kecil nilai NLL, semakin baik model dalam memprediksi distribusi data[49]. NLL dihitung dengan rumus berikut:

$$i(\theta) = -\sum_{i=1}^{n} (y_i log \hat{y}_{\theta,i} + (1 - y_i) \log (1 - \hat{y}_{\theta,i})) \quad (2,3)$$

$i(\theta)$: Fungsi *negative log likelihood* yang merupakan fungsi dari parameter vektor θ\thetaθ.

$\sum_{i-1}^{n}$: Penjumlahan semua data i dari 1 sampai n

y_i : Variabel biner yang bernilai 1 jika kejadian yang diinginkan terjadi dan 0 jika tidak.

$\hat{y}_{\theta,i}$: Probabilitas yang diprediksi (di bawah model yang diparameterkan oleh θ\thetaθ) bahwa kejadian ke-i terjadi (yaitu, $y_i = 1$).

2.2.12 *Confusion Matrix*

Confusion matrix merupakan sebuah tabel yang menunjukkan kinerja sebuah algoritma, terutama seberapa baik performanya dalam melakukan klasifikasi[50]. Berikut merupakan tabel dari *confusion matrix.*

Tabel 2.2 Tabel *confusion matrix*

		Kelas yang sebenarnya	
Kelas yang diprediksi		*Positive*	*Negative*
	Positive	*True positive (TP)*	*False negative (FN)*
	Negative	*False positive (FP)*	*True negative (TN)*

Berdasarkan tabel *confusion matrix*, dapat dihasilkan perhitungan matrix seperti *accuracy, precision, recall,* dan juga *f1-score.*

a. *Accuracy*

Accuracy merupakan persentase prediksi model yang benar. *Accuracy* dihitung dengan menggunakan rumus berikut:

$$accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \quad (2.4)$$

b. *Precision*

Precision merupakan persentase prediksi positif yang benar. *Precision* dihitung dengan rumus berikut:

$$precision = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \quad (2.5)$$

c. *Recall*

Recall merupakan persentase daru kelas positif yang benar-benar terdeteksi oleh model. *Recall* memiliki rumus sebagai berikut:

$$recall = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \quad (2.6)$$

d. *F1-Score*

F1-score merupakan gabungan dari kombinasi antara precision dan recall. *F1-score* dihitung dengan rumus berikut:

$$F1 = \frac{2 \times precision \times recall}{precisionn + recall} \quad (2.7)$$

2.2.13 *Mean Average Precision (mAP)*

mAP (*Mean Average Precision*) dihitung dengan mengambil rata-rata nilai *precision* untuk berbagai tingkat *recall*, dari 0 hingga 1. Langkah pertama adalah melakukan tinjauan singkat mengenai *precision*, *recall*, dan IoU. *Precision* mengukur seberapa akurat prediksi terhadap suatu objek, sedangkan *recall* menilai seberapa baik sistem menemukan semua kelas positif. [51]. mAP dirumuskan sebagai berikut:

$$mAP = \frac{1}{n} \sum_{recall} precision(recall_i) \quad (2.8)$$

N = banyaknya nilai *recall*

mAP = average precision

2.2.14 PyTorch

PyTorch merupakan *library* dalam bahasa python yang memudahkan dalam pembuatan proyek *deep learning*. PyTorch menekankan fleksibilitas dan memungkinkan model *deep learning* untuk diekspresikan dalam gaya penulisan Python secara umum. Melalui pendekatan ini, pytorch menarik minat dari komunitas riset sejak pertama kali dirilis[52].

2.2.15 *OpenCV*

OpenCV (Open Source Computer Vision) merupakan *library* perangkat lunak khusus yang digunakan untuk *computer vision*[53]. *Library* ini pertama kali diluncurkan pada 1999 oleh intel. Pada awalanya, *library* ini dibuat untuk memajukan intensif CPU[54]. OpenCV dapat dipakai untuk berbagai bahasa pemrograman seperti C++, Java, Perl, dan juga Python[55].

2.2.16 Streamlit

Streamlit adalah sebuah *framework* berbasis Python yang bersifat *open source*. *Framework* ini dirancang untuk memudahkan pengembang dalam membuat aplikasi web interaktif di bidang *data science* dan *machine learning*. Salah satu keunggulan Streamlit adalah pengembang tidak perlu mengatur tampilan situs web menggunakan CSS, HTML, atau JavaScript, karena *framework* Streamlit sudah menyediakan fungsi-fungsi yang mengurus aspek-aspek tersebut[56].