

bab i

pendahuluan

1.1 latar belakang

Indonesia, dengan posisinya sebagai negara tropis, diberkahi dengan kondisi geografis yang sangat kondusif yang memungkinkan para petani untuk mengembangkan berbagai jenis tanaman. Faktor iklim di Indonesia sangat mendukung untuk aktivitas pertanian yang bisa dilakukan sepanjang tahun tanpa henti. Ini berarti, Indonesia memiliki keanekaragaman tanaman pertanian yang luas, yang tidak hanya memperkaya keragaman hayati, tetapi juga menjadi sumber utama pangan bagi penduduknya. Dari padi, jagung, sampai tanaman hortikultura dan perkebunan, beragamnya iklim dan tanah di Indonesia berkontribusi pada keberhasilan pertanian yang menjadi tulang punggung ekonomi dan kehidupan sosial masyarakat. Kondisi geografis yang beragam, dari dataran rendah yang subur hingga dataran tinggi dengan iklim yang sejuk, semakin menambah potensi Indonesia sebagai lumbung pangan yang mampu memproduksi berbagai jenis tanaman sepanjang tahun (Husodo, dkk, 2004).

Produksi jagung dunia menempati urutan ke tiga setelah padi dan gandum yaitu 612,5 juta ton. Di Indonesia jagung berada di posisi kedua, tepat setelah padi dan berperan penting dalam ekonomi, dikarenakan kegunaannya yang beragam, termasuk sebagai sumber makanan, bahan pakan, dan bahan dasar untuk industri. Distribusi penanaman jagung terus meluas di berbagai negara di dunia karena tanaman jagung mempunyai daya adaptasi yang luas di daerah subtropis ataupun tropis. (Rukmana, 2008). Jagung, sebagai tanaman semusim atau annual, menyelesaikan seluruh siklus hidupnya, yang berlangsung antara 80 hingga 150 hari, dalam satu periode tanam saja. Proses ini dibagi menjadi dua fase utama yang krusial untuk pengembangan dan produksi tanaman. Paruh pertama dari siklus hidup jagung didedikasikan untuk tahap pertumbuhan vegetatif, dimana tanaman secara aktif mengembangkan struktur fisiknya seperti akar, batang, dan daun. Selama fase ini, jagung menumpuk biomassa dan mempersiapkan diri untuk fase selanjutnya dengan memaksimalkan penyerapan nutrisi dan fotosintesis untuk mendukung pertumbuhannya. (Megasari & Nuriyati, n.d.)

Jagung memiliki beragam manfaat, di mana buahnya menjadi sumber utama karbohidrat untuk konsumsi manusia. Daunnya bermanfaat sebagai pakan untuk ternak seperti kambing, sapi, dan kerbau. Selain itu, batang dan jenggel jagung yang telah kering dapat dimanfaatkan sebagai bahan bakar. Kulit buah jagung juga memiliki kegunaan alternatif, seperti pengganti kertas rokok atau sebagai pembungkus untuk makanan ringan seperti dodol. Buah jagung sendiri dapat diolah menjadi aneka jenis makanan, termasuk nasi jagung, jagung bakar, berondong jagung (popcorn), serta dijadikan pakan ternak. (Mubyarto, 2002).

Pada tahun 2017, Indonesia menjadi produsen jagung terbesar di dunia menurut data dari organisasi pertanian dunia (FAO). Namun, faktor penyakit pada jagung mengakibatkan rendahnya produksi jagung (Muhammad Imron Rosadi dkk, 2021). Penyakit adalah kondisi yang disebabkan oleh mikroorganisme patogen, seperti jamur, bakteri, atau virus, yang menginfeksi tanaman dan menyebabkan kerusakan atau gangguan pada proses fisiologis tanaman. Sedangkan hama adalah organisme, seperti

serangga atau hewan kecil lainnya, yang merusak tanaman dengan cara memakan, menghisap, atau merusak bagian tanaman secara fisik. hama dapat menyebarkan penyakit dengan membawa atau mentransfer patogen, seperti virus, bakteri, atau jamur, dari satu tanaman ke tanaman lainnya saat mereka menghisap atau memakan bagian tanaman yang terinfeksi. (hamidson & suwandi, 2019).

terdapat 3 penyakit utama pada daun jagung yaitu karat daun, bercak daun, dan hawar daun. penyakit karat daun pada jagung ditandai oleh bercak merah kekuningan (mirip karat besi). bercak daun ditandai dengan bercak kecoklatan atau kekuningan. sedangkan hawar daun ditandai dengan bercak oval kecil yang memanjang menjadi bentuk elips dan akhirnya menjadi besar (rosadi & lutfi, 2021). infeksi penyakit ini dapat mengakibatkan penurunan produksi, kerugian ekonomi, dan bahkan kegagalan panen jika tidak dikelola dengan baik. tidak semua petani memiliki kemampuan untuk mengidentifikasi pola tersebut. terkadang, petani terlambat menyadari bahwa tanaman mereka mengalami penyakit setelah sebelumnya menganggap tanaman tersebut dalam kondisi sehat. namun, setelah diamati oleh dinas pertanian, tanaman tersebut terindikasi mengalami penyakit. keadaan ini dapat menyebabkan kerugian, terutama jika mengakibatkan gagal panen (robby kurniawan budhi dkk, 2019).

dalam upaya mengidentifikasi penyakit pada daun tanaman jagung, petani selama ini melakukan pengamatan secara manual. namun, dengan kemajuan teknologi informasi yang pesat saat ini, terdapat peluang untuk mengatasi berbagai masalah umum yang dihadapi, terutama bagi para petani. hal ini menjadi tantangan bagi para pengembang teknologi informasi untuk menciptakan sistem yang dapat menjadi solusi bagi masalah yang dialami oleh para petani (rosadi & lutfi, 2021).

penelitian ini dilakukan dengan tujuan utama untuk mengembangkan sebuah sistem yang dapat mendeteksi penyakit-penyakit yang mungkin menyerang tanaman jagung. metode yang digunakan dalam sistem ini adalah *faster r-cnn* sebuah algoritma *deep learning* yang telah terbukti efektif dalam pengenalan pola pada citra. dengan adanya sistem ini, diharapkan petani atau ahli pertanian dapat dengan cepat dan akurat mengidentifikasi penyakit-penyakit yang mengancam tanaman jagung, sehingga langkah-langkah penanganan yang tepat dapat segera dilakukan untuk mencegah kerusakan yang lebih parah dan potensi kegagalan panen.

1.2 teori

1.2.1 jagung

jagung dianggap sebagai salah satu elemen kunci dalam sistem pangan indonesia, menempati posisi sebagai makanan utama bagi sejumlah besar penduduk di berbagai wilayah, seperti provinsi jawa tengah, jawa timur, pulau madura, beberapa daerah di sulawesi, dan di nusa tenggara timur. jagung berfungsi sebagai komponen esensial dalam pakan untuk sektor peternakan, khususnya unggas, yang mana daging ayam dan telur yang dihasilkan menjadi sumber protein nabati yang terjangkau untuk kebutuhan konsumsi masyarakat. jagung juga digunakan sebagai bahan dasar dalam industri pengolahan pangan, yang meliputi pembuatan tepung jagung, berbagai jenis roti dan kue, hingga snack. (suryana & agustian, 2014)



gambar 1 tanaman jagung

jagung mengandung karbohidrat kompleks yang menjadi sumber energi utama bagi tubuh. karbohidrat ini dicerna secara perlahan, sehingga memberikan energi yang bertahan lebih lama. jagung mengandung serat yang baik untuk pencernaan. serat membantu melancarkan buang air besar dan mencegah sembelit. jagung mengandung berbagai vitamin seperti vitamin b1 (tiamin), b3 (niasin), b5 (asam pantotenat), dan vitamin c. selain itu, jagung juga mengandung mineral penting seperti magnesium, fosfor, dan besi. jagung kaya akan antioksidan seperti lutein dan zeaxanthin yang baik untuk kesehatan mata. antioksidan ini membantu melindungi mata dari kerusakan yang disebabkan oleh sinar uv dan radikal bebas. kandungan serat dalam jagung memberikan efek kenyang lebih lama, yang dapat membantu dalam pengelolaan berat badan dengan mengurangi keinginan untuk ngemil. dengan berbagai manfaat ini, jagung tidak hanya menjadi makanan yang enak tetapi juga memberikan kontribusi penting bagi kesehatan tubuh.

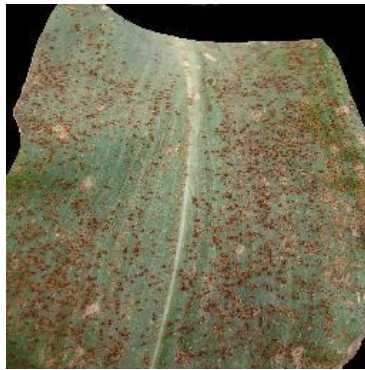
1.2.2 penyakit pada daun jagung

penyakit merupakan suatu keadaan di mana sel-sel dan jaringan-jaringan pada tanaman mengalami disfungsi atau tidak dapat beroperasi secara normal. kondisi ini disebabkan oleh faktor-faktor lingkungan atau oleh agen patogen, yang berujung pada penggambaran gejala tertentu (agrios, 2005). penyakit pada tanaman adalah situasi di mana tanaman mengalami kerusakan atau gangguan yang bukan disebabkan oleh serangan hama, tetapi oleh virus, bakteri, atau jamur. penyakit jagung merupakan masalah serius dalam pertanian yang dapat mengurangi hasil panen dan kualitas jagung.

gejala penyakit bisa bervariasi, tetapi sering melibatkan perubahan warna dan tekstur daun, penurunan pertumbuhan tanaman, hingga kematian tanaman. berikut adalah beberapa jenis penyakit yang sering menyerang daun tanaman jagung:

1. karat daun (*puccinia polysora*)

karat daun pada tanaman jagung disebabkan oleh cendawan *puccinia polysora*. gejala awalnya muncul berupa bercak berwarna oranye hingga coklat tua pada permukaan daun. seiring perkembangannya, bercak ini dapat membesar dan menyatu, membentuk area yang terinfeksi. jika tidak ditangani dengan tepat, karat daun dapat menyebabkan kerusakan serius pada daun dan mengganggu proses fotosintesis. (hamidson & suwandi, 2019)



gambar 2 penyakit karat daun

2. hawar daun (*helminthosporium maydis*)

hawar daun adalah penyakit yang disebabkan oleh cendawan *helminthosporium maydis*. gejala awal dari penyakit ini biasanya mulai terlihat pada tanaman berusia sekitar dua minggu, dengan daun bagian bawah menjadi target awal serangan penyakit ini. pada tahap awal, muncul bercak-bercak berbentuk oval berwarna hijau tua atau hijau kelabu. seiring waktu, bercak-bercak ini berkembang menjadi lesi yang lebih besar dan menyebabkan daun mengering yang dapat merusak dan membunuh daun sepenuhnya. (hamidson & suwandi, 2019).



gambar 3 penyakit hawar daun

3. bercak daun (*bipolaris maydis*)

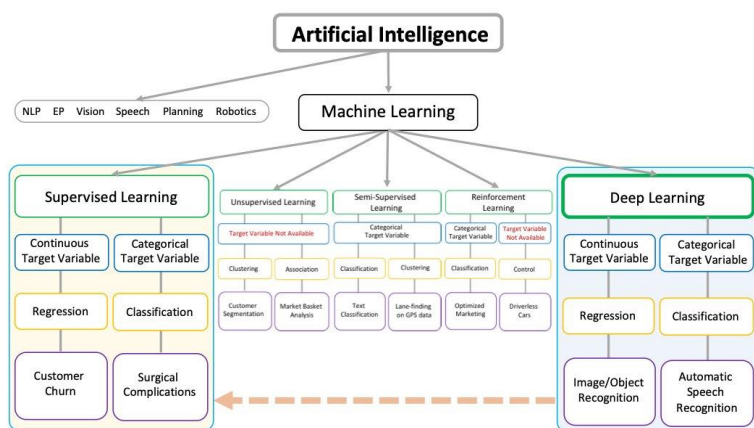
penyakit bercak daun pada tanaman jagung disebabkan oleh cendawan *bipolaris maydis* . gejalanya meliputi munculnya bercak berbentuk bulat atau oval dengan tengah abu-abu dan tepi coklat tua pada daun. bercak ini dapat menggabung dan membentuk area yang lebih besar. penyakit ini dapat mengakibatkan penurunan area daun yang berfungsi untuk fotosintesis. (agrotan, 2015)



gambar 4 penyakit bercak daun

1.2.3 kecerdasann buatan (*artificial intelligence / ai*)

kecerdasan buatan (ai) adalah cabang ilmu komputer yang fokus pada pengembangan sistem yang dapat meniru atau bahkan melampaui tugas-tugas yang biasanya memerlukan kecerdasan manusia. ai memungkinkan komputer untuk memahami, belajar, dan membuat keputusan secara mandiri, dengan mengintegrasikan prinsip matematika, logika, dan psikologi kognitif. ai tidak hanya meniru kemampuan manusia, tetapi juga dapat melakukan tugas yang melibatkan analisis data besar dengan kecepatan dan akurasi yang lebih tinggi. teknologi ini berpotensi merevolusi berbagai sektor, seperti industri, pendidikan, dan kesehatan, serta meningkatkan efisiensi dalam kehidupan dan pekerjaan kita (muhammad dahria, 2008).



gambar 5 hierarki kecerdasan buatan

1.2.4 machine learning

pembelajaran mesin (ml) adalah teknik utama dalam bidang kecerdasan buatan (ai), yang dirancang untuk mensimulasikan dan mengadopsi perilaku belajar manusia untuk mengatasi berbagai tantangan atau memperkenalkan otomatisasi. inti dari ml adalah kemampuannya untuk memahami dan meniru cara belajar dan pengambilan keputusan oleh manusia atau entitas cerdas lainnya, memungkinkan sistem untuk mengenal pola dan membuat generalisasi. dua aplikasi fundamental dari ml meliputi klasifikasi, di mana data dikategorikan ke dalam grup-grup berbeda, dan prediksi, yang memanfaatkan data historis untuk memprediksi kejadian di masa depan. proses pembelajaran ini sangat bergantung pada 'data pelatihan', sebuah set data khusus yang digunakan untuk mengajarkan algoritma ml bagaimana menginterpretasi dan merespons input baru. ini menekankan pentingnya proses pelatihan dalam mengembangkan model ml yang efektif dan akurat (ahmad, 2017).

1.2.5 deep learning

deep learning merupakan subdisiplin dari *machine learning* yang memanfaatkan algoritma berlapis untuk analisis data yang kompleks. berbeda dari metode regresi sederhana, teknik ini menciptakan representasi data yang kompleks dan dinamis, memungkinkan model untuk mengenali pola dari berbagai faktor melalui algoritma non-linier. ini memfasilitasi pemahaman yang lebih mendalam tentang data, meningkatkan kemampuan prediksi dan klasifikasi. ketika algoritma pembelajaran mesin diberi akses ke dataset yang luas selama proses pelatihan, ia mulai mengenali pola dan hubungan antar elemen seperti bentuk, warna, dan kata. karena kemampuan ini, sistem dapat diterapkan untuk membuat prediksi yang akurat berdasarkan data yang dianalisis (pakpahan, 2021).

1.2.6 computer vision

computer vision, sebagai cabang penting dari artificial intelligence, fokus pada pemberian kemampuan kepada komputer untuk menginterpretasikan dan memahami dunia visual seperti yang dilakukan manusia. melalui penggunaan gambar dari kamera digital atau video, serta model-model canggih berbasis *deep learning*, *computer vision* mampu mengidentifikasi, mengklasifikasikan, dan memahami objek dalam gambar atau video. tujuan utama dari teknologi ai-vision ini adalah untuk mengotomatisasi berbagai tugas visual yang biasanya dilakukan oleh manusia, meningkatkan efisiensi dan akurasi dalam berbagai aplikasi praktis, dari pengenalan wajah hingga navigasi otomatis (utama, 2020).

1.2.7 klasifikasi

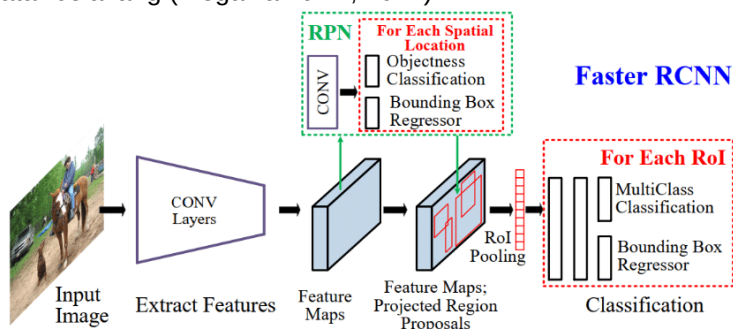
klasifikasi adalah metode yang membangun model atau fungsi untuk mengelompokkan data ke dalam kategori yang ditentukan, bertujuan untuk mengidentifikasi kelas objek yang belum diklasifikasikan. proses ini terbagi menjadi dua tahap: pertama, tahap pembelajaran atau pelatihan, di mana model dikembangkan dari data latih; kedua, tahap pengujian, di mana model yang telah dilatih diuji dengan data baru untuk mengevaluasi kinerjanya (azizah, 2023).

1.2.8 pengolahan citra digital

pengolahan citra digital merupakan teknik mengolah informasi dengan menggunakan citra sebagai masukan dan menghasilkan citra atau sebagian dari citra sebagai keluaran. proses ini bertujuan untuk meningkatkan kualitas citra sehingga lebih mudah dipahami oleh manusia atau diproses oleh computer (raysyah dkk., 2021).

1.2.9 faster r-cnn

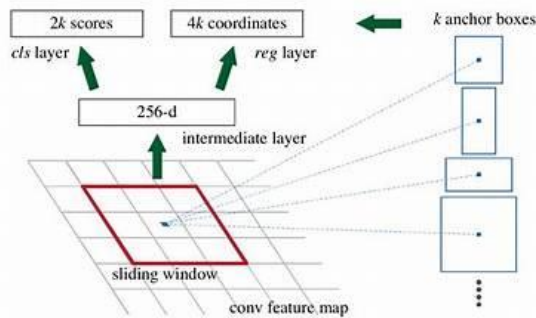
faster r-cnn adalah metode yang mempercepat deteksi objek dengan memanfaatkan gabungan dari dua teknik lain: *cnn*, untuk memproses gambar, dan *rpn*, untuk menemukan area penting dalam gambar. ini dimulai dengan mengolah gambar melalui beberapa lapisan untuk membuat peta fitur, kemudian menentukan area spesifik yang mungkin mengandung objek. teknik ini membagi hasilnya menjadi dua kategori: objek dan latar belakang, membuatnya lebih cepat dan efisien dalam mengenali objek dalam gambar (megawan dkk., 2022). pada bagian lapisan *roi pooling*, gambar dibagi menjadi wilayah berukuran tertentu misalnya 7x7. kemudian, *max-pooling* diterapkan untuk memilih nilai maksimum dari setiap sel dalam wilayah tersebut. proses konvolusi juga dilakukan pada setiap piksel. hasil dari kedua proses ini kemudian digabungkan kembali di lapisan *roi pooling* untuk membedakan mana yang merupakan objek dan mana yang latar belakang (megawan dkk., 2022).



gambar 6 arsitektur *faster r-cnn*

faster r-cnn terbagi menjadi dua komponen utama yaitu:

1. *region proposal network* (rpn) yang efektif dalam mengidentifikasi lokasi potensial objek dalam gambar secara cepat. lokasi objek pada gambar ditentukan oleh area yang dikenal sebagai region of interest (roi). roi diperkenalkan oleh ross girschik sebagai metode deteksi objek menggunakan pendekatan *deep learning*. layer yang menggunakan roi memanfaatkan feature maps yang merupakan hasil dari *cnn* yang melibatkan berbagai convolution layers dan *max pooling* layers. sebuah matriks $n \times n$ dibentuk dengan membagi ruang feature map menjadi roi, di mana kolom pertama adalah indeks dari gambar dan kolom-kolom lainnya adalah koordinat dari roi mulai dari koordinat atas kiri sampai bawah kiri. roi yang telah ditentukan ini dikenal sebagai region proposal (megawan dkk., 2022).



gambar 7 proses rpn

2. *classifier* adalah proses yang digunakan untuk mengategorikan roi yang telah diidentifikasi oleh rpn ke dalam kelas atau target yang relevan. pada tahap ini, teknik yang digunakan adalah cnn (megawan dkk., 2022).

peningkatan kecepatan: *faster r-cnn* menawarkan peningkatan substansial dalam kecepatan deteksi objek melalui penggunaan *region proposal network* (rpn) yang terintegrasi, memungkinkan pembuatan proposal wilayah secara lebih cepat dan efisien. berikut adalah beberapa kelebihan *faster r-cnn*:

1. efisiensi komputasi:
integrasi rpn dalam *faster r-cnn* meminimalisir jumlah proposal yang harus diproses, yang berkontribusi pada pengurangan signifikan dalam beban komputasi serta mempercepat proses pelatihan dan inferensi.
2. akurasi yang lebih tinggi:
model ini menunjukkan peningkatan akurasi dalam mendeteksi dan mengklasifikasikan objek berkat proses integrasi fitur dari lapisan konvolusional dengan proposal wilayah yang akurat, meningkatkan kemampuan model dalam pengidentifikasian objek.
3. kemampuan mengatasi variasi ukuran objek:
dengan menggunakan kotak jangkar (*anchor boxes*) yang beragam dalam hal rasio dan skala, *faster r-cnn* efektif dalam mendeteksi objek dengan berbagai ukuran dan bentuk, suatu aspek penting untuk aplikasi praktis yang luas.
4. pelatihan secara *end-to-end*:
keseluruhan jaringan, dari rpn hingga lapisan klasifikasi dan regresi, dapat dilatih secara *end-to-end*, memfasilitasi penyetelan halus semua parameter jaringan secara simultan.

1.2.10 lapisan konvolusi

lapisan konvolusi adalah elemen kunci dalam struktur arsitektur cnn, di mana ia mengimplementasikan operasi konvolusi terhadap output dari lapisan sebelumnya. proses konvolusi, yang merupakan konsep matematis, melibatkan penerapan fungsi yang berulang kali diaplikasikan pada hasil dari fungsi lain. dalam konteks ini, operasi konvolusi memanfaatkan dua fungsi yang memiliki nilai nyata untuk menghasilkan sebuah peta fitur yang mewakili gambar input. output dan input ini dianggap sebagai parameter dengan nilai nyata. selain itu, ukuran dari output yang dihasilkan oleh lapisan

ini dapat diatur melalui penggunaan hyperparameters. proses ini adalah bagian penting yang membantu dalam pembentukan arsitektur jaringan cnn, memungkinkan jaringan untuk mempelajari dan mengekstraksi fitur penting dari gambar yang diberikan (iswantoro & un, 2022). persamaan (1) berikut merupakan persamaan yang digunakan untuk menghitung nilai yang diharapkan:

$$(W - F + 2P)/(S + 1) \quad (1)$$

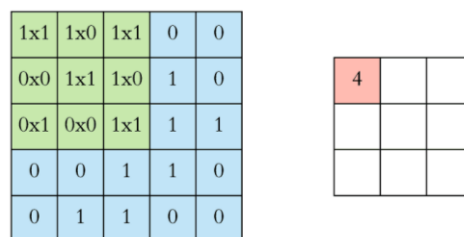
keterangan :

w = ukuran volume gambar;

f = ukuran filter;

p = nilai padding yang digunakan;

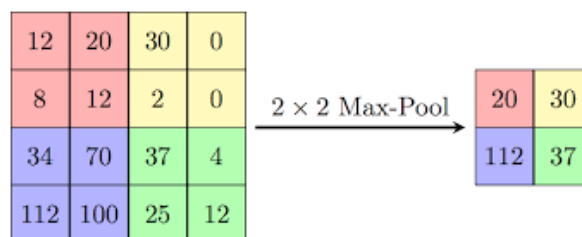
s = ukuran pergeseran (stride)



gambar 8 konvolusi layer

1.2.11 pooling layer

pooling merupakan proses yang bertujuan untuk mengurangi dimensi matriks melalui operasi yang disebut *pooling*. lapisan *pooling* biasanya diletakkan setelah lapisan konvolusi dalam arsitektur jaringan. lapisan ini menggunakan filter dengan ukuran dan stride yang spesifik untuk bergerak secara sistematis melintasi peta fitur. cara kerja yang umum dalam lapisan *pooling* adalah menggunakan filter berukuran 2x2 yang diterapkan setiap dua langkah, mengolah setiap segmen dari input yang diberikan. operasi ini efektif untuk mengkondensasi informasi penting sambil mengurangi jumlah data yang harus diproses oleh jaringan (iswantoro & un, 2022).



gambar 9 max-pooling layer

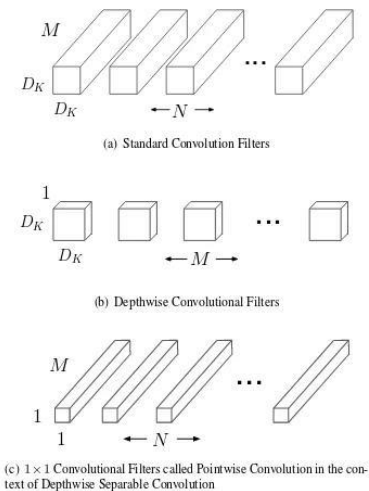
1.2.12 lapisan fully-connected

lapisan *fully-connected* adalah sebuah lapisan di mana setiap neuron dari lapisan sebelumnya terhubung dengan semua neuron di lapisan berikutnya, mirip dengan struktur jaringan saraf umum. perbedaan utama antara lapisan *fully-connected*

dan lapisan konvolusi terletak pada cara koneksi neuron. dalam lapisan konvolusi, setiap neuron hanya terhubung ke area lokal dari input, sedangkan di lapisan *fully-connected*, setiap neuron terhubung secara keseluruhan dengan lapisan lainnya. meskipun kedua jenis lapisan ini menggunakan produk titik dalam operasinya, fungsi mereka tidak terlalu berbeda satu sama lain (iswanto & un, 2022).

1.2.13 mobilenet

mobilenets adalah arsitektur jaringan saraf konvolusional (cnn) yang dirancang oleh para peneliti google untuk mengurangi penggunaan sumber daya komputasi yang berlebihan. sesuai dengan namanya, "mobile", arsitektur ini dikembangkan khusus untuk perangkat seluler. perbedaan utama antara mobilenet dan cnn tradisional terletak pada penggunaan lapisan konvolusi yang menyesuaikan ketebalan filter dengan ketebalan gambar masukan. mobilenet memecah proses konvolusi menjadi dua bagian: konvolusi *depthwise* dan konvolusi *pointwise* (rizqi, 2018).



gambar 10 konvolusi standard (a) dibagi menjadi dua lapisan: *depthwise convolution* (b) dan *pointwise convolution* (c) untuk membuat filter terpisah secara mendalam (depthwise).



gambar 1 kiri: lapisan konvolusi standard dengan *batchnorm* dan *relu*. kanan: *depthwise convolution* dan *pointwise convolution* dengan *batchnorm* dan *relu*.

Table 1. MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32 \text{ dw}$	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64 \text{ dw}$	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128 \text{ dw}$	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128 \text{ dw}$	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256 \text{ dw}$	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256 \text{ dw}$	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
5x	Conv dw / s1	$3 \times 3 \times 512 \text{ dw}$
	Conv / s1	$1 \times 1 \times 512 \times 512$
	Conv dw / s2	$3 \times 3 \times 512 \text{ dw}$
	Conv / s1	$1 \times 1 \times 512 \times 1024$
	Conv dw / s2	$3 \times 3 \times 1024 \text{ dw}$
	Conv / s1	$1 \times 1 \times 1024 \times 1024$
	Avg Pool / s1	Pool 7×7
	FC / s1	1024×1000
	Softmax / s1	Classifier

gambar 12 arsitektur mobilenet

1.2.14 stochastic gradient descent

sgd atau *stochastic gradient descent* adalah sebuah algoritma optimasi yang bertujuan untuk menemukan bobot baru dengan cara memilih secara acak satu data dari keseluruhan data pelatihan. setelah pemilihan data, sgd menganalisis data tersebut. penggunaan sgd dapat meminimalkan kebutuhan memori selama proses pemrosesan bobot baru dan mempercepat proses pembelajaran model. namun, proses pembelajaran model ini cenderung mengalami fluktuasi yang signifikan (ruder, 2016). berikut ini adalah urutan proses algoritma sgd:

1. menambah t pada setiap iterasi menggunakan persamaan (2) berikut.

$$t=t+1 \quad (2)$$

2. menghitung nilai gradien menggunakan persamaan (3) berikut.

$$g_t = \frac{\delta L}{\delta W} \quad (3)$$

3. menghitung weight average menggunakan persamaan (4) berikut.

$$v_t = \gamma \cdot v_t + \eta \cdot g_t \quad (4)$$

4. memperbarui parameter menggunakan persamaan (5) berikut.

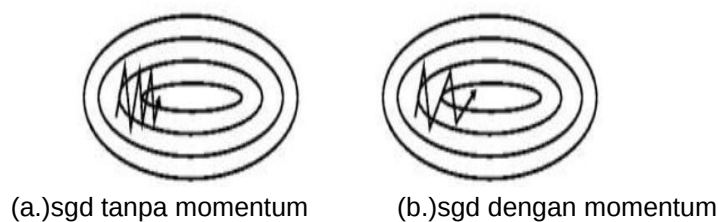
$$\theta = \theta_t - v_t \quad (5)$$

keterangan :

t = iterasi

g = gradien
 δ = turunan fungsi loss
 δw = turunan weight
 v = weight average
 γ = momentum
 η = learning rate
 θ = parameter yang di perbaiki

momentum adalah teknik yang digunakan untuk mempercepat sgd dalam arah yang tepat dan mengurangi osilasi. teknik ini bekerja dengan menambahkan parameter momentum γ dari vektor pembaruan sebelumnya ke vektor pembaruan yang sedang berlangsung (ruder, 2016).



gambar 13 gambar sgd

1. 2.15 *confusion matrix multiclass*

confusion matrix adalah representasi visual yang mengkategorikan hasil prediksi model klasifikasi berdasarkan nilai sebenarnya atau target dari data uji, memberikan gambaran jelas tentang performa model dalam menangani berbagai kelas dalam klasifikasi. matriks ini membagi prediksi ke dalam empat segmen esensial: *true positives* (tp), *true negatives* (tn), *false positives* (fp), dan *false negatives* (fn), yang memungkinkan analisis mendetail tentang jumlah dan jenis kesalahan yang dibuat oleh model. ini sangat penting karena beberapa jenis kesalahan mungkin memiliki konsekuensi yang lebih serius tergantung pada konteks aplikasi (indriani, 2014). *confusion matrix* juga memfasilitasi penghitungan metrik evaluasi seperti *accuracy*, *precision*, *recall*, dan *f1-score*, yang menyediakan wawasan komprehensif tentang kemampuan model untuk menangani data secara efisien dan efektif, menjadikan alat diagnostik yang kuat untuk mengoptimalkan model dan mengurangi jenis kesalahan yang paling merugikan dalam praktik sebenarnya.

1. akurasi

akurasi adalah hasil akurasi untuk mendefinisikan tingkat dari dekatnya nilai aktual dengan nilai hasil prediksi. semakin tinggi tingkat akurasi, semakin baik kinerja metode. adapun rumus untuk melakukan perhitungan akurasi seperti pada persamaan (6) berikut:

$$Akurasi = \frac{TP}{tp + fp + fn} \quad (6)$$

2. *precision*

precision adalah tingkat keakuratan data yang diminta dengan data yang diberikan oleh model. *precision* merupakan klasifikasi tp dari keseluruhan data yang diprediksi positif menjadi nilai kelas positif. adapun rumus untuk melakukan perhitungan *precision* seperti pada persamaan (7) berikut:

$$Precision = \frac{TP}{tp + fp} \quad (7)$$

3. *recall*

recall adalah tingkat dari keberhasilan dari sistem untuk mendapatkan informasi. adapun rumus untuk melakukan perhitungan *recall* seperti pada persamaan (8) berikut:

$$recall = \frac{TP}{tp + fn} \quad (8)$$

4. *f1-score*

f1-score yang merupakan perbandingan *precision* dan *recall*. adapun rumus untuk melakukan perhitungan *f1 score* seperti pada persamaan (9) berikut:

$$f1 - score = \frac{2 \times precision \times recall}{precision + recall} \quad (9)$$

1.3 rumusan masalah

berdasarkan latar belakang yang telah dijelaskan sebelumnya, penelitian ini bertujuan untuk menjawab beberapa rumusan masalah berikut ini:

1. bagaimana membuat model deteksi penyakit daun tanaman jagung menggunakan metode *faster r-cnn*?
2. bagaimana membangun sistem deteksi penyakit daun tanaman jagung?

1.4 tujuan penelitian

dalam penelitian ini, terdapat beberapa tujuan yang ingin dicapai, antara lain:

1. membangun model deteksi penyakit daun tanaman jagung menggunakan metode *faster r-cnn*.
2. membangun sistem deteksi penyakit daun pada tanaman jagung.

1.5 manfaat penelitian

adapun manfaat dari penelitian ini adalah sebagai berikut :

1. tersedia system deteksi penyakit pada daun tanaman jagung yang diharapkan dapat digunakan bagi petani.
2. hasil dari penelitian diharapkan dapat menjadi referensi bagi peneliti selanjutnya.

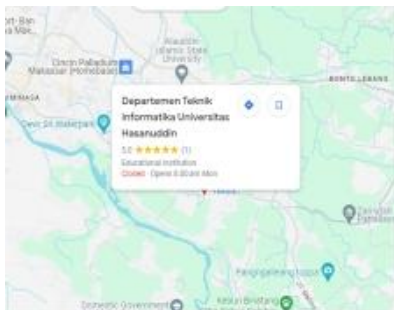
1.6 ruang lingkup/asumsi perancangan

penelitian ini membatasi hanya 4 class penyakit pada daun tanaman jagung yaitu: bercak daun, karat daun, hawar daun, dan daun sehat.

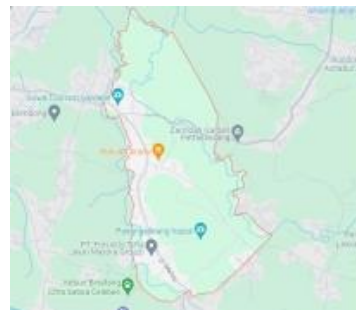
bab 2 metode penelitian/perancangan

2.1 waktu dan lokasi penelitian

penelitian skripsi ini dilakukan di 2 tempat. pengumpulan dataset dilakukan di desa mata allo (kebun jagung). pengolahan data dilakukan di laboratorium seis departemen teknik informatika. waktu penelitian ini dimulai sejak proposal penelitian ini disetujui hingga selesai.



departemen teknik informatika
universitas hasanuddin



desa mata allo

gambar 14 lokasi penelitian

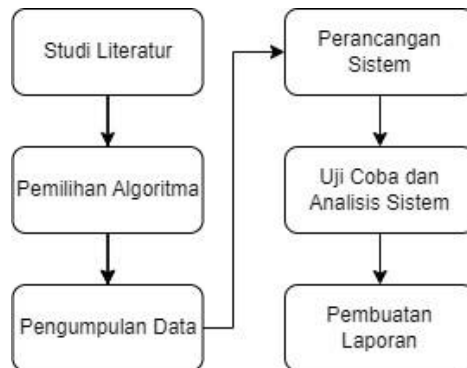
2.2 instrumen penelitian

dalam penelitian ini, instrumen yang digunakan meliputi komponen perangkat lunak dan perangkat keras sebagai berikut:

- a. software :
 - 1. visual studio code
 - 2. roboflow
 - 3. google chrome
 - 4. google collaboratory
 - 5. microsoft (word, excel, & ppt)
 - 6. google drive
- b. hardware :
 - 1. laptop hp (ram 12gb, ssd)
 - 2. printer
 - 3. wi-fi (jaringan internet)

2.3 tahapan penelitian

penelitian ini dilakukan secara terstruktur dan bertahap sesuai dengan diagram tahapan penelitian yang ditampilkan pada gambar di bawah.



gambar 15 tahapan penelitian

a. studi literatur

tahap ini melibatkan pencarian dan analisis literatur terkait dengan topik penelitian, mencakup penyakit daun jagung, teknik *deep learning*, dan sistem sebelumnya dalam deteksi penyakit tanaman. studi literatur membantu peneliti memahami konteks dan kerangka kerja yang telah ada dalam domain tersebut, serta membantu mengidentifikasi kelemahan dan peluang yang masih perlu dieksplorasi. selain itu, penelitian literatur juga memberikan pemahaman yang mendalam tentang metode dan algoritma *deep learning* yang relevan untuk penelitian ini.

b. pemilihan algoritma

berdasarkan studi literatur yang dilakukan, teknik *deep learning* yang akan digunakan adalah *faster r-cnn*. keputusan ini didasarkan pada fakta bahwa *faster r-cnn* telah terbukti memberikan hasil yang optimal dalam tugas deteksi objek dalam gambar, termasuk dalam deteksi penyakit pada daun tanaman jagung. dengan kemampuan untuk mengenali pola-pola kompleks dalam data gambar dan tingkat kecepatan yang lebih tinggi, *faster r-cnn* diharapkan dapat memberikan hasil yang baik dalam mengidentifikasi penyakit daun tanaman jagung dengan efisien.

c. pengumpulan data

pengumpulan data dilakukan di kebun jagung secara langsung dengan cara mengambil gambar jenis-jenis penyakit yang ada pada daun tanaman jagung. langkah ini melibatkan kunjungan ke lapangan, pengamatan langsung, dan pengambilan gambar secara rinci untuk mengumpulkan data primer yang akan menjadi dasar utama dalam pelatihan dan pengujian model deteksi penyakit jagung menggunakan *faster r-cnn*. kedua, sebagai pendukung, data sekunder dari sumber seperti kaggle juga digunakan. data sekunder ini dapat mencakup dataset gambar penyakit jagung yang sudah ada dan diunggah oleh berbagai sumber di kaggle. pemanfaatan data sekunder dapat memperluas cakupan dataset, meningkatkan keberagaman penyakit yang dapat dideteksi.

d. perancangan sistem

dalam tahap ini, peneliti merancang konsep dan alur program, kemudian mengimplementasikannya sesuai dengan rancangan yang telah dibuat. model

akan dibangun menggunakan bahasa *python* dan dengan memanfaatkan beberapa library, dengan fokus pada penggunaan algoritma *faster r-cnn*.

e. uji coba dan analisis sistem

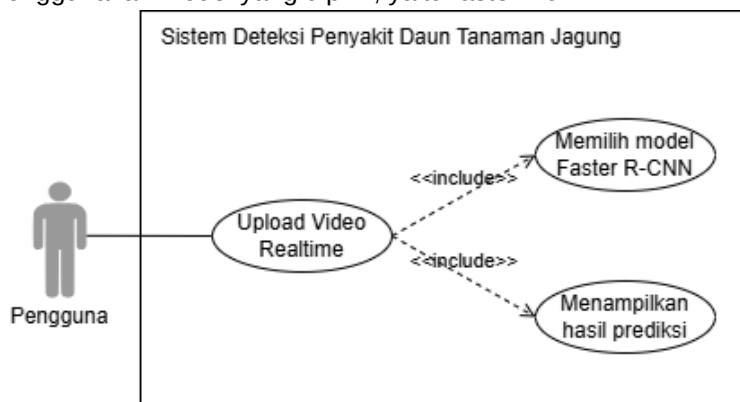
pada tahap ini, penelitian akan menguji efektivitas model deteksi penyakit daun tanaman jagung yang telah dibangun dengan menggunakan cv2 untuk deteksi secara real-time. proses uji coba akan mencakup mengambil sejumlah sampel gambar jagung yang memiliki berbagai jenis penyakit dan menggunakan model yang telah dihasilkan untuk melakukan deteksi secara langsung pada gambar-gambar tersebut. hasil dari uji coba ini akan direkam dan dianalisis dengan seksama. analisis sistem ini akan mencakup evaluasi kinerja model, termasuk tingkat akurasi, sensitivitas, dan spesifisitas dalam mendeteksi penyakit daun tanaman jagung secara real-time.

f. pembuatan laporan

pada tahap ini, peneliti akan menyusun laporan final yang mendokumentasikan seluruh rangkaian kegiatan penelitian dari awal hingga akhir. laporan penelitian akan mencakup beberapa elemen kunci, termasuk pendahuluan, yang menjelaskan latar belakang, tujuan, dan ruang lingkup penelitian. selanjutnya, metodologi yang digunakan dalam penelitian akan dijelaskan secara rinci, termasuk langkah-langkah yang diambil dalam pengumpulan data, pengembangan model, dan proses analisis sistem.

2.4 diagram use case

gambar 16 menunjukkan diagram use case untuk sistem deteksi penyakit daun tanaman jagung yang dirancang untuk mendeteksi jenis penyakit pada daun jagung secara real-time. diagram ini menggambarkan interaksi antara pengguna dengan sistem, mulai dari mengupload video secara real-time hingga menampilkan hasil prediksi menggunakan model yang dipilih, yaitu *faster r-cnn*.



gambar 16 diagram use case sistem deteksi penyakit daun jagung

2.5 teknik pengumpulan data

a. data primer

peneliti mengumpulkan data primer dengan mengunjungi kebun jagung lalu mengambil gambar pada objek penyakit daun tanaman jagung secara langsung. gambar 17 menunjukkan teknik pengumpulan data menggunakan handphone.



gambar 17 teknik pengumpulan data primer

b. data sekunder

sumber data sekunder ini berupa dataset gambar penyakit daun tanaman jagung yang sudah ada dalam bentuk digital di kaggle, yang kemudian digunakan sebagai tambahan dalam penelitian deteksi penyakit daun tanaman jagung. berikut adalah contoh data yang berhasil dikumpulkan.



gambar 18 contoh data

2.6 preprocessing data

dataset yang digunakan dalam penelitian ini memiliki berbagai variasi ukuran gambar, seperti 900x900, 1200x1200, 1200x800, dan lainnya. untuk memastikan data pelatihan memiliki kualitas yang seragam dan konsisten, setiap gambar dalam dataset akan melalui proses penyesuaian ukuran (resize) sehingga memiliki dimensi yang sama. ukuran yang dipilih untuk semua gambar adalah 640x640 piksel. langkah ini bertujuan agar proses pelatihan model faster r-cnn dapat berjalan lebih efektif dan hasil deteksi lebih akurat.



gambar sebelum di resize

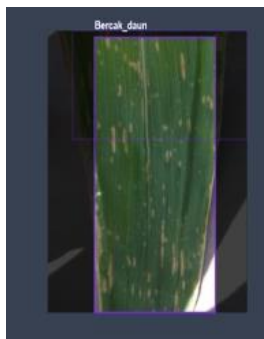


gambar setelah di resize

gambar 19 contoh data proses resize

2.7 pelabelan data

penelitian ini menggunakan *roboflow* untuk melakukan pelabelan data. data dibagi menjadi 4 kelas yaitu, hawar daun, bercak daun, karat daun dan daun sehat.



gambar pelabelan data
kelas bercak daun



gambar pelabelan data
kelas daun sehat



gambar pelabelan data
kelas hawar daun



gambar pelabelan data
kelas karat daun

gambar 20 pelabelan data

2.8 pembagian data

setelah proses pelabelan data selesai, langkah berikutnya adalah membagi dataset menjadi tiga bagian, yaitu data pelatihan (train) 80%, data pengujian (test) 10%, dan data validasi 10%. data pelatihan digunakan untuk melatih model, data pengujian digunakan untuk mengukur kinerja model, sementara data validasi digunakan untuk menyesuaikan parameter model dan mencegah overfitting.

2.9 data augmentasi

tujuan dari augmentasi data adalah untuk memperkaya dataset pelatihan dengan variasi yang lebih besar, sehingga model dapat memahami dan mengenali objek dengan lebih baik, bahkan dalam situasi yang berbeda-beda. dengan menggunakan augmentasi data seperti ini, model pembelajaran mesin dapat menjadi lebih robust dan mampu menghadapi variasi dalam data yang sebenarnya saat digunakan dalam pengujian atau pemakaian praktis. penelitian ini akan menggunakan 3 teknik augmentasi data yaitu flip dan rotation.

a. flip horizontal dan vertikal

pada tahap ini, gambar-gambar dalam dataset dapat diubah secara horizontal atau vertikal. flip horizontal mengacu pada pemutaran gambar sepanjang sumbu vertikal, seperti melihat gambar dari arah berlawanan, sedangkan flip vertikal adalah pemutaran sepanjang sumbu horizontal, menghasilkan gambar yang terbalik secara atas-bawah.



data sebelum flip



data setelah flip horizontal



data setelah flip vertikal

gambar 21 contoh proses flip

b. rotation

data augmentasi juga melibatkan rotasi gambar. rotasi dilakukan dengan memutar gambar dalam kisaran antara -15° hingga $+15^{\circ}$. ini memungkinkan model untuk belajar dari variasi sudut pandang yang berbeda terhadap objek dalam gambar.



data sebelum rotasi

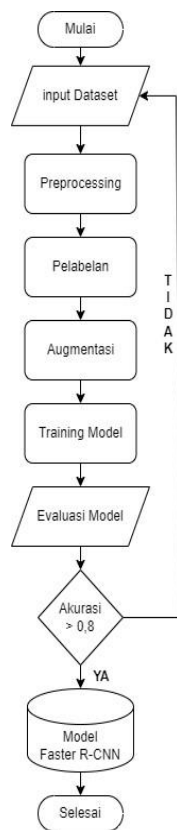


data setelah rotasi

gambar 22 contoh proses rotasi

2.10 perancangan dan implementasi sistem

setelah data telah di augmentasi langkah selanjutnya adalah melatih model menggunakan algoritma *faster r-cnn* (*faster region-based convolutional neural network*). berikut adalah flowchart rancangan pelatihan model.



gambar 23 flowchart tahap train, test dan validasi

a. import library

tahap awal pembuatan model adalah mengimport library yang digunakan. gambar 24 menunjukkan beberapa library yang diperlukan untuk proses pelatihan.

```

from torchvision import datasets, models
from torchvision.transforms import functional as FT
from torchvision import transforms as T
from torch import nn, optim
from torch.nn import functional as F
from torch.utils.data import DataLoader, sampler, random_split, Dataset
import numpy as np
import pandas as pd
import os
import torch
import torchvision
import copy
import math
from PIL import Image
import cv2
import sys
import math
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from tqdm import tqdm
from pycocotools.coco import COCO
from pycocotools.cocoeval import COCOeval
  
```

gambar 24 import library

b. import dataset from *roboflow*

dataset yang telah dipersiapkan dari *roboflow*, selanjutnya di import lagi. gambar 25 menunjukkan source code untuk melakukan import dataset dari *roboflow*.

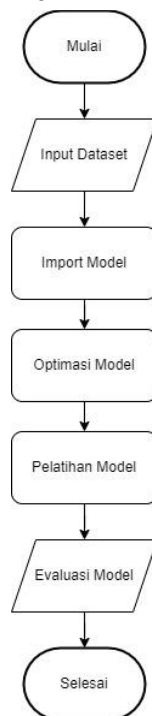
```
!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="PrwwRHdU4TQCAu8mv58n")
project = rf.workspace("skripsi-alifa").project("deteksi-9xwuj")
dataset = project.version(1).download("coco")
```

gambar 25 import dataset from *roboflow*

c. build model

pada tahap ini ada 3 langkah utama yang dilakukan. gambar 26 menunjukkan alur dari tahapan pembuatan model *faster r-cnn*.



gambar 26 tahap pembuatan model *faster r-cnn*

1. import model *faster r-cnn*

gambar menunjukkan *source code* yang digunakan untuk membuat model *faster r-cnn* yang digunakan dalam penelitian ini.

```
model = models.detection.fasterrcnn_mobilenet_v3_large_fpn(pretrained=True)
in_features = model.roi_heads.box_predictor.cls_score.in_features
model.roi_heads.box_predictor = models.detection.faster_rcnn.FastRCNNPredictor(in_features, n_classes)
```

gambar 27 load model *faster r-cnn*

2. model optimizer

```
params = [p for p in model.parameters() if p.requires_grad]
optimizer = torch.optim.SGD(params, lr=0.0001, momentum=0.9, nesterov=True, weight_decay=1e-4)
```

gambar 28 source code optimasi model

gambar menunjukkan *source code* yang digunakan untuk optimasi model *faster r-cnn*. pertama, kita mengambil parameter-parameter (*weights dan biases*) dari model yang memerlukan pembelajaran (*requires_grad*). hal ini dilakukan dengan menggunakan *list comprehension* untuk menyaring parameter-parameter yang bersifat *trainable*. selanjutnya, kita menginisialisasi optimizer menggunakan metode *stochastic gradient descent* (sgd) dengan konfigurasi sebagai berikut:

2.1. *lr=0.0001*: ini adalah tingkat pembelajaran (*learning rate*) yang digunakan oleh algoritma sgd. tingkat pembelajaran mengontrol seberapa besar perubahan yang diterapkan pada parameter model pada setiap langkah pembelajaran. nilai yang kecil seperti 0.0001 menunjukkan perubahan yang sangat kecil pada setiap iterasi.

2.2. *momentum=0.9*: momentum 0.9 mengindikasikan bahwa sebagian besar dari perubahan yang dilakukan pada setiap iterasi adalah hasil dari perubahan pada iterasi sebelumnya.

2.3. *nesterov=true*: ketika *nesterov=true*, algoritma akan "mengintip" ke depan untuk melihat perubahan yang mungkin terjadi pada langkah selanjutnya.

2.4 *weight decay=1e-4*: ini adalah teknik regularisasi yang membantu mencegah *overfitting*. *weight decay* (penalization pada parameter-parameter) dengan nilai 1e-4 mengindikasikan bahwa kita memberikan penghukuman yang kecil pada parameter untuk mendorong model agar tidak terlalu kompleks.

d. training model

```
def train_one_epoch(model, optimizer, loader, device, epoch):
    model.to(device)
    model.train()
    all_losses = []
    all_losses_dict = []
    for images, targets in tqdm(loader):
        images = list(image.to(device) for image in images)
        targets = [{k: torch.tensor(v).to(device) for k, v in t.items()} for t in targets]
        loss_dict = model(images, targets)
        losses = sum(loss for loss in loss_dict.values())
        loss_dict_append = {k: v.item() for k, v in loss_dict.items()}
        loss_value = losses.item()
        all_losses.append(loss_value)
        all_losses_dict.append(loss_dict_append)
        if not math.isfinite(loss_value):
            print(f'Loss is {loss_value}, stopping training')
            print(loss_dict)
            sys.exit(1)
        optimizer.zero_grad()
        losses.backward()
        optimizer.step()
    all_losses_dict = pd.DataFrame(all_losses_dict)
```

gambar 29 source code training model

gambar menampilkan *source code* yang digunakan pada penelitian ini untuk melatih model *faster r-cnn*. fungsi `train_one_epoch` dalam kode ini dirancang untuk melatih model selama satu *epoch*. fungsi ini memulai dengan mengirim model ke perangkat yang ditentukan (seperti cpu atau gpu). selama proses pelatihan, model berada dalam mode pelatihan (`model.train()`), yang memungkinkan fitur-fitur seperti *dropout* dan *batch normalization* untuk berfungsi dalam mode pelatihan. fungsi ini kemudian mengiterasi melalui setiap *batch* dari data *loader*. untuk setiap *batch*, gambar dan target diubah agar kompatibel dengan perangkat (misalnya, dipindahkan ke gpu jika menggunakan cuda). model kemudian diaplikasikan pada *batch* tersebut untuk menghasilkan dictionary kerugian (`loss_dict`). total kerugian dihitung dengan menjumlahkan semua kerugian individual dari *dictionary* ini. selanjutnya, dilakukan pemeriksaan untuk memastikan nilai kerugian adalah *finite*; jika tidak, pelatihan dihentikan. *optimizer* mengatur *gradien* menjadi nol (`.zero_grad()`), lalu melakukan *backpropagation* dari kerugian (`losses.backward()`) dan langkah optimisasi selanjutnya (`optimizer.step()`). informasi kerugian dari setiap *batch* disimpan dalam list, yang pada akhirnya dikonversi menjadi *dataframe pandas*, memberikan gambaran detail tentang kerugian selama *epoch* tersebut. fungsi ini efektif untuk pelatihan model dalam *setting deep learning*, memberikan *feedback* terperinci tentang proses dan kemajuan pelatihan.

e. model evaluate

gambar menampilkan source code yang digunakan untuk menampilkan evaluasi dari model yang sudah latih.

```
epoch_losses.append(np.mean(all_losses))
epoch_loss_classifier.append(all_losses_dict['loss_classifier'].mean())
epoch_loss_box.append(all_losses_dict['loss_box_reg'].mean())
epoch_loss_rpn_box.append(all_losses_dict['loss_rpn_box_reg'].mean())
epoch_loss_object.append(all_losses_dict['loss_objectness'].mean())

print("Epoch {}, lr: {:.6f}, loss: {:.6f}, loss_classifier: {:.6f}, loss_box: {:.6f}, loss_rpn_box: {:.6f}, loss_object: {:.6f}".format(
    epoch, optimizer.param_groups[0]['lr'], np.mean(all_losses),
    all_losses_dict['loss_classifier'].mean(),
    all_losses_dict['loss_box_reg'].mean(),
    all_losses_dict['loss_rpn_box_reg'].mean(),
    all_losses_dict['loss_objectness'].mean()
))
```

gambar 30 source code menampilkan evaluasi model

kode diatas bertujuan untuk menampilkan metrik kinerja selama proses pelatihan pada setiap *epoch*. di dalamnya, nilai rata-rata dari kerugian total (*epoch_losses*) dan kerugian spesifik dari berbagai komponen model - seperti kerugian klasifikasi (*epoch_loss_classifier*), kerugian kotak pembatas (*epoch_loss_box*), kerugian kotak pembatas dari *region proposal network* (*epoch_loss_rpn_box*), dan kerugian objektivitas (*epoch_loss_object*) - dihitung dari data yang terkumpul selama *epoch*. nilai-nilai ini dihitung sebagai rata-rata dari kerugian yang terkumpul dalam *all_losses_dict*, yang merupakan *dictionary* yang menyimpan detail kerugian dari setiap iterasi dalam *epoch* tersebut. selain itu digunakan juga metrik evaluasi seperti; akurasi, *precision*, *recall*, dan *f1-score* untuk evaluasi model. hal ini memastikan bahwa performa model dalam mendeteksi penyakit daun jagung dinilai secara komprehensif, baik dari segi ketepatan prediksi maupun kemampuan model dalam mengidentifikasi seluruh kategori penyakit secara efektif.

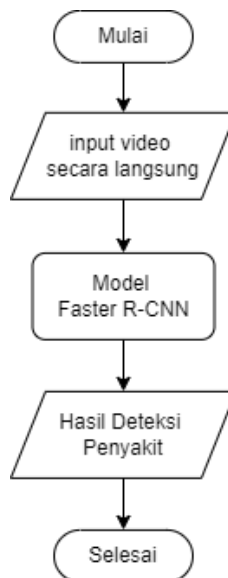
f. convert model to .pth

untuk melakukan save model ke file .pth, digunakan kode yang ditampilkan pada gambar 31.

```
torch.save(model.state_dict(), 'skripsi-alifa.pth')
```

gambar 31 source code save model

g. rancangan sistem deteksi secara *realtime*



gambar 32 rancangan sistem deteksi secara realtime

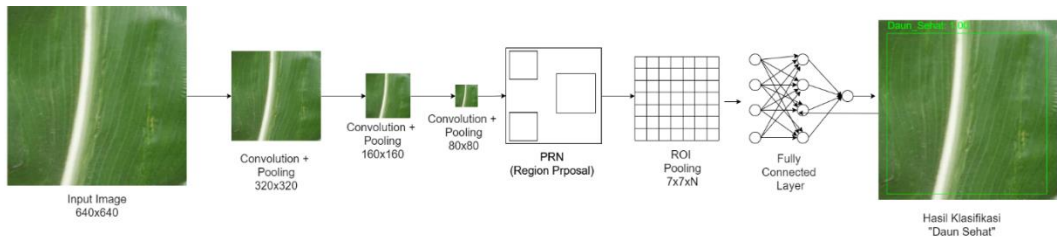
berikut adalah penjelasan alur sistem deteksi penyakit pada daun jagung menggunakan model faster r-cnn, sesuai dengan gambar:

1. mulai : proses dimulai dengan menjalankan sistem deteksi.
2. input video secara langsung : sistem menerima input berupa video langsung (real-time) yang menunjukkan daun jagung. video ini akan diproses secara langsung untuk mendeteksi adanya penyakit pada daun.
3. model faster r-cnn : video input kemudian diproses oleh model faster r-cnn. model ini berfungsi untuk mengenali dan mendeteksi pola atau ciri-ciri penyakit pada daun jagung dalam setiap frame video. model telah dilatih sebelumnya untuk mendeteksi berbagai jenis penyakit pada daun jagung.
4. hasil deteksi penyakit : setelah video diproses, hasil deteksi berupa informasi tentang keberadaan penyakit, jenis penyakit, dan lokasi penyakit pada daun jagung ditampilkan.
5. selesai : proses selesai setelah hasil deteksi ditampilkan dan sistem dapat siap untuk mendeteksi frame video berikutnya.

alur ini memungkinkan deteksi penyakit pada daun jagung secara otomatis dan real-time, sehingga dapat memudahkan pemantauan kondisi kesehatan tanaman.

2.11 tahapan *faster r-cnn*

gambar 33 menunjukkan alur kerja model *faster r-cnn*, mulai dari pengolahan gambar masukan hingga menghasilkan prediksi akhir berupa label kelas dan posisi *bounding box*.

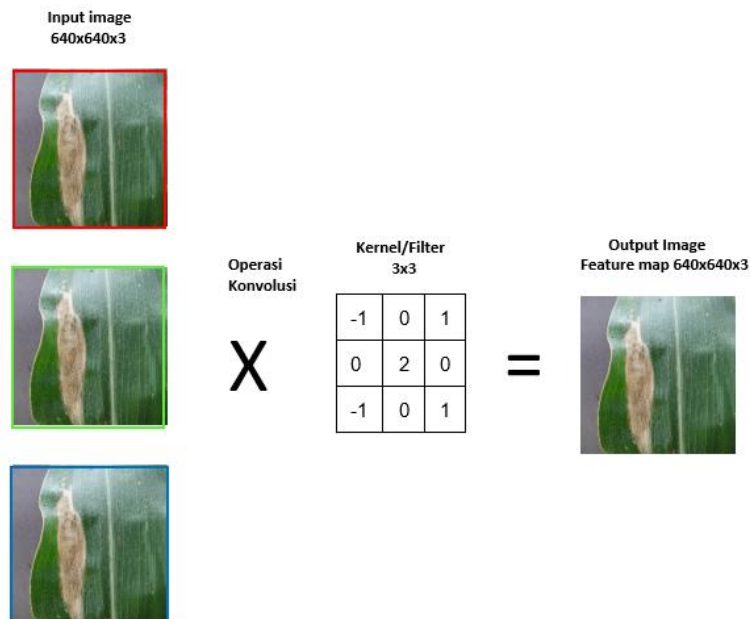


gambar 33 tahapan *faster r-cnn*

model *faster r-cnn* yang digunakan diinisialisasi dengan arsitektur *mobilenet v3*, yang telah dilatih sebelumnya (*pretrained=true*). konfigurasi ini disesuaikan untuk mendeteksi penyakit daun jagung. langkah-langkah utama dalam model *faster r-cnn* terdiri dari tiga bagian utama: lapisan konvolusi, *region proposal network* (rpn), dan klasifikasi serta penyesuaian *bounding box*.

1. lapisan konvolusi

pada tahap awal, gambar masukan dengan ukuran 640x640 piksel diproses melalui lapisan konvolusi. tujuannya adalah untuk mengekstraksi fitur penting dari gambar menggunakan filter yang dilatih. tahap ini merupakan langkah awal dari lima lapisan dalam arsitektur *faster r-cnn*. pada proses ini, feature map yang dihasilkan memiliki dimensi 640x640 dengan tiga channel utama: merah (red), hijau (green), dan biru (blue). gambar input yang terdiri dari tiga channel tersebut akan mengalami proses konvolusi menggunakan kernel berukuran 3x3 dengan stride 1. hasil dari tahap ini adalah feature map baru yang diperoleh dari operasi konvolusi pada citra input.



gambar 34 ilustrasi proses konvolusi pada data citra daun jagung

proses:

- a) lapisan konvolusi menghasilkan peta fitur (*feature map*) yang mewakili pola atau struktur tertentu, seperti tepian atau tekstur.
- b) lapisan pooling digunakan untuk mengurangi dimensi peta fitur, mengurangi kebutuhan memori, dan mempercepat perhitungan.

1.1 proses konvolusi channel red

tahap ini merupakan langkah awal dari lima lapisan dalam arsitektur fsater r-cnn. pada proses ini, feature map yang dihasilkan memiliki dimensi 640×640 dengan tiga channel utama: merah (red), hijau (green), dan biru (blue). gambar input yang terdiri dari tiga channel tersebut akan mengalami proses konvolusi menggunakan kernel berukuran 3×3 dengan stride 1. hasil dari tahap ini adalah feature map baru yang diperoleh dari operasi konvolusi pada citra input.

	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	AS	AT	AU	AV	AW	AX	AY	AZ	BA	BB	BC	BD	BE	BF	BG	BH	BI	BJ	BK	BL	BM	
1	32	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	
2	122	123	119	111	107	107	104	105	109	111	110	111	114	114	109	110	119	106	104	107	113	120	120	119	119	125	123	119	119	113	117	135	
3	122	121	116	122	120	126	127	120	112	109	107	100	109	109	100	100	100	106	104	107	113	119	119	117	118	123	129	119	120	116	115	126	
4	120	120	119	110	112	127	122	121	110	112	111	106	107	100	107	101	105	104	107	112	119	118	116	120	124	114	119	116	115	123	123		
5	119	119	119	121	125	130	134	123	113	113	115	112	110	112	110	110	119	108	107	110	114	119	119	117	117	119	121	108	117	117	116	122	
6	121	120	119	120	123	126	129	119	113	115	118	112	110	115	115	122	126	116	115	116	118	121	121	120	120	121	110	120	120	121	116	121	
7	126	122	119	110	119	119	120	121	117	118	118	111	111	119	119	121	126	116	115	116	117	119	119	114	122	120	114	122	120	114	116	121	
8	121	125	120	119	117	116	116	123	120	121	120	112	112	120	122	122	125	119	117	116	115	115	115	116	117	117	117	115	121	117	111	113	
9	134	126	121	119	119	117	122	121	123	123	115	115	123	123	120	121	121	119	116	114	115	117	118	117	114	119	118	115	119	115	116	115	
10	133	126	123	123	123	122	121	120	124	126	120	121	127	123	116	114	126	126	124	121	119	119	121	122	122	120	114	118	117	115	122	121	
11	122	120	121	134	129	121	118	121	122	122	122	122	122	122	120	120	124	125	127	127	125	122	119	118	116	115	113	115	117	119	122	121	
12	131	129	133	130	128	123	123	126	122	123	124	124	124	122	122	124	125	127	127	125	122	119	117	116	115	117	129	122	123	123	123	121	
13	137	134	132	126	123	129	123	123	125	126	127	127	126	125	123	123	126	127	127	126	124	121	119	118	118	118	119	120	121	124	124	124	122
14	127	122	120	128	128	121	120	124	126	126	129	131	132	132	131	130	129	127	129	129	127	125	123	120	119	120	122	125	120	127	124	122	121
15	125	121	120	133	132	132	134	128	128	130	132	133	133	132	131	130	130	129	129	127	124	122	122	123	125	127	129	126	122	121	121	121	121
16	134	130	134	134	131	131	124	131	130	132	132	132	133	134	134	133	133	133	131	130	128	126	124	124	126	128	129	128	125	122	122	124	124
17	140	138	141	137	133	131	125	132	132	132	132	132	133	134	135	135	135	135	135	135	135	135	135	135	135	135	135	135	135	135	135	135	135
18	140	141	146	143	137	134	125	133	132	132	132	132	133	134	135	137	137	137	137	137	137	137	137	137	137	137	137	137	137	137	137	137	137
19	136	139	149	150	143	137	135	134	134	133	132	132	132	132	134	136	138	138	138	138	138	138	138	138	138	138	138	138	138	138	138	138	138
20	140	142	150	150	149	141	135	138	138	137	137	137	138	139	139	139	139	139	139	139	139	139	139	139	139	139	139	139	139	139	139	139	139
21	144	146	149	151	141	141	141	142	143	144	143	141	138	135	133	133	134	135	136	135	134	132	131	130	131	129	128	124	122	122	122	122	122
22	149	149	149	149	149	149	149	149	149	149	149	149	149	149	149	149	149	149	149	149	149	149	149	149	149	149	149	149	149	149	149	149	149
23	152	151	149	146	145	146	147	150	150	150	150	149	148	144	139	134	131	128	140	141	140	139	136	134	132	133	132	132	132	132	131	129	129
24	149	147	145	145	145	146	149	154	153	151	149	147	145	143	140	137	135	133	133	133	140	141	142	141	138	136	132	131	133	135	135	131	131
25	148	145	145	146	147	149	149	157	155	152	149	147	146	145	144	142	141	141	140	140	140	143	145	144	141	138	132	130	133	135	135	131	131
26	147	145	145	147	149	151	152	157	155	151	148	146	146	147	147	146	144	143	143	143	144	145	144	141	137	134	133	134	135	134	135	134	131
27	148	147	146	148	151	153	154	154	151	149	146	145	146	147	148	148	148	148	148	148	148	148	148	148	148	148	148	148	148	148	148	148	148
28	149	148	148	149	152	154	155	149	147	145	144	145	146	148	150	151	149	147	144	143	145	147	146	142	139	138	138	141	145	139	132	136	134
29	145	149	147	150	151	150	145	144	142	144	145	146	148	150	151	149	147	146	145	147	146	144	141	140	140	141	142	143	139	139	141	146	146
30	142	147	150	153	156	159	162	160	148	147	147	148	149	151	151	151	150	149	148	146	144	143	142	141	140	141	142	143	139	139	141	146	146
31	142	145	154	155	153	156	162	160	148	147	147	148	149	151	151	151	150	149	148	146	144	143	142	141	140	141	142	143	139	139	141	146	146

gambar 35 matriks intensitas piksel citra daun jagung 640x640

karena matriks 640x640 mempunyai proses konvolusi yang sangat panjang, maka kali ini akan disederhanakan dengan konvolusi citra daun jagung berukuran 9x9.

red	1	2	3	4	5	6	7	8	9
1	145	120	116	122	121	122	131	145	188
2	146	148	140	131	125	123	147	159	142
3	147	149	149	145	144	163	166	149	160
4	145	152	152	154	173	168	151	149	147
5	140	155	156	175	177	153	150	144	139
6	136	156	166	182	162	152	141	128	123
7	151	164	200	159	162	145	134	124	118
8	159	201	161	174	170	165	165	157	139
9	196	162	164	162	166	159	157	143	159

filter 3x3		
-1	0	1
0	2	0
-1	0	1

x

red	1	2	3	4	5	6	7
1	269	278	262	270	266	268	282
2	256	272	263	275	271	274	290
3	252	261	270	282	276	280	292
4	242	258	280	291	283	278	289
5	230	245	269	276	272	269	279
6	220	233	261	268	274	260	271
7	218	228	256	263	258	253	264

=

red	1	2	3	4	5	6	7	8	9
1	145	120	116	122	121	122	131	145	188
2	146	148	140	131	125	123	147	159	142
3	147	149	149	145	144	163	166	149	160
4	145	152	152	154	173	168	151	149	147
5	140	155	156	175	177	153	150	144	139
6	136	156	166	182	162	152	141	128	123
7	151	164	200	159	162	145	134	124	118
8	159	201	161	174	170	165	165	157	139
9	196	162	164	162	166	159	157	143	159

red.	1	2	3	4	5	6	7	8	9		filter 3x3		red.	1	2	3	4	5	6	7			
1	145	120	116	122	121	122	131	145	188	x	-1	0	1	=	1	269	278	262	270	266	268	282	
2	146	148	140	131	125	123	147	159	142		0	2	0	2	2	256	272	263	275	271	274	290	
3	147	149	149	145	144	163	166	149	160		-1	0	1	3	3	252	261	270	282	276	280	292	
4	145	152	152	154	173	168	151	149	147		4	242	258	280	291	4	242	258	280	291	283	278	289
5	140	155	156	175	177	153	150	144	139		5	230	245	269	276	5	230	245	269	276	272	269	279
6	136	156	166	182	162	152	141	128	123		6	220	233	261	268	6	220	233	261	268	274	260	271
7	151	164	200	159	162	145	134	124	118		7	218	228	256	263	7	218	228	256	263	258	253	264
8	159	201	161	174	170	165	165	157	139														
9	196	162	164	162	166	159	157	143	159														

gambar 36 proses konvolusi channel red

proses konvolusi yang terjadi menggunakan filter berukuran 3x3 dengan stride 1 secara matematis adalah sebagai berikut:

tabel 1 perhitungan konvolusi

posisi sub-matriks	perhitungan konvolusi	hasil
(1,1)	$(145 \times -1) + (120 \times 0) + (116 \times 1) + (146 \times 0) + (148 \times 2) + (140 \times 0) + (147 \times -1) + (149 \times 0) + (149 \times 1)$ $= (-145) + (0) + (116) + (0) + (296) + (0) + (-147) + (0) + (149)$	269
(1,2)	$(120 \times -1) + (116 \times 0) + (122 \times 1) + (148 \times 0) + (140 \times 2) + (131 \times 0) + (149 \times -1) + (149 \times 0) + (145 \times 1)$ $= (-120) + (0) + (122) + (0) + (280) + (0) + (-149) + (0) + (145)$	278
(1,3)	$(116 \times -1) + (122 \times 0) + (121 \times 1) + (140 \times 0) + (131 \times 2) + (125 \times 0) + (149 \times -1) + (145 \times 0) + (144 \times 1)$ $= (-116) + (0) + (121) + (0) + (262) + (0) + (-149) + (0) + (144)$	262

proses ini akan berlanjut hingga seluruh citra selesai diproses melalui konvolusi, menghasilkan feature map berukuran 7x7.

1.2 feature map channel green

proses konvolusi pada channel green dilakukan dengan cara yang sama seperti pada channel red dengan menggunakan filter yang sama. berikut adalah hasil dari konvolusi channel green yang menghasilkan feature map berukuran 7x7.

tabel 2 feature map channel green

green	1	2	3	4	5	6	7
1	207	219	224	230	226	228	244
2	199	210	217	228	222	225	240
3	195	206	219	233	229	234	247
4	185	200	225	236	230	226	238
5	175	187	215	224	220	218	228
6	168	179	206	215	212	209	219
7	163	174	200	208	205	202	212

1.3 feature map channel blue

proses konvolusi pada channel blue dilakukan dengan cara yang sama seperti pada channel red dengan menggunakan filter yang sama. berikut adalah hasil dari konvolusi channel green yang menghasilkan feature map berukuran 7x7.

tabel 3 feature map channel blue

blue	1	2	3	4	5	6	7
1	189	200	186	194	188	192	205
2	180	194	188	182	168	159	154
3	186	187	200	205	195	185	180
4	194	199	212	214	202	191	188
5	188	193	207	210	198	187	183
6	192	198	211	205	195	183	179
7	205	210	223	216	204	193	189

menggabung feature map channel red, green, dan blue sehingga menjadi multi-channel.

tabel 4 multi-channel feature map

posisi (i,j)	red	green	blue
(1,1)	269	207	189
(1,2)	278	219	200
(1,3)	262	224	186
(1,4)	270	230	194
(1,5)	266	226	188
(1,6)	268	228	192
(1,7)	282	224	205

penggabungan feature map dari red, green, dan blue menghasilkan multi-channel feature map yang memberikan representasi yang lebih lengkap dan kaya tentang gambar.

hasil:

gambar 640x640 dilakukan pooling secara bertahap menjadi peta fitur berukuran 320x320, 160x160, dan akhirnya 80x80, seperti terlihat pada bagian awal gambar. peta fitur terakhir ini digunakan oleh rpn untuk mengidentifikasi lokasi objek.

2. region proposal network (rpn)

region proposal network (rpn) bekerja di atas peta fitur terakhir untuk menghasilkan proposal wilayah (*region proposals*) yang berpotensi mengandung objek. proses:

- rpn memindai peta fitur terakhir dan menentukan area mana yang paling mungkin mengandung objek.

- b) menggunakan *anchor box* dengan 3 skala berbeda dan 3 aspek rasio, rpn memberikan *bounding box* awal beserta skor probabilitas untuk masing-masing wilayah.

menghitung pergeseran koordinat untuk bounding box menggunakan regresi linier dengan 4 parameter:

δx = pergeseran horizontal

δy = pergeseran vertical

δw = perubahan lebar

δh = perubahan tinggi

tabel 5 region proposal

proposal	koordinat (x,y,w,h)	skor
box 1	(55, 47, 138, 123)	0.92
box 2	(60, 50, 140, 120)	0.90
box 3	(65, 55, 135, 118)	0.85
box 4	(58, 52, 137, 122)	0.82
box 5	(70, 60, 132, 115)	0.78

hasil:

proposal wilayah dengan kemungkinan tertinggi digunakan untuk langkah selanjutnya, yaitu klasifikasi dan penyesuaian *bounding box*, seperti yang ditunjukkan pada bagian "rpn" pada gambar 35.

3. klasifikasi dan penyesuaian bounding box

pada tahap akhir, proposal wilayah dari rpn diteruskan ke roi pooling untuk menyeragamkan ukuran menjadi 7x7 sebelum dimasukkan ke lapisan sepenuhnya terhubung (*fully connected layer*).

proses:

- a) roi pooling mengonversi setiap *region proposal* ke dimensi tetap, memungkinkan pengolahan yang konsisten.

bounding box awal dari rpn:

$$(x,y,w,h) = (55,47,138,123) \quad (10)$$

menghitung perbaikannya:

$$\delta x = 5, \delta y = 3, \delta w = 10, \delta h = 5 \quad (11)$$

bounding box setelah penyesuaian:

$$\begin{aligned} x' &= x + \delta x = 55 + 5 = 60 \\ y' &= y + \delta y = 47 - 3 = 44 \\ w' &= w + \delta w = 138 + 10 = 148 \\ h' &= h + \delta h = 123 - 5 = 118 \end{aligned} \quad (12)$$

tabel 6 region proposal tetap

proposal	koordinat (x,y,w,h)	skor klasifikasi	iou
box 1	(60, 44, 148, 118)	0.92	1.00
box 2	(65, 46, 145, 120)	0.90	0.85
box 3	(70, 50, 140, 110)	0.78	0.75

- b) jaringan sepenuhnya terhubung (*fully connected*) melakukan klasifikasi untuk menentukan jenis objek, dalam hal ini terdeteksi sebagai "daun sehat".

tabel 7 hasil klasifikasi

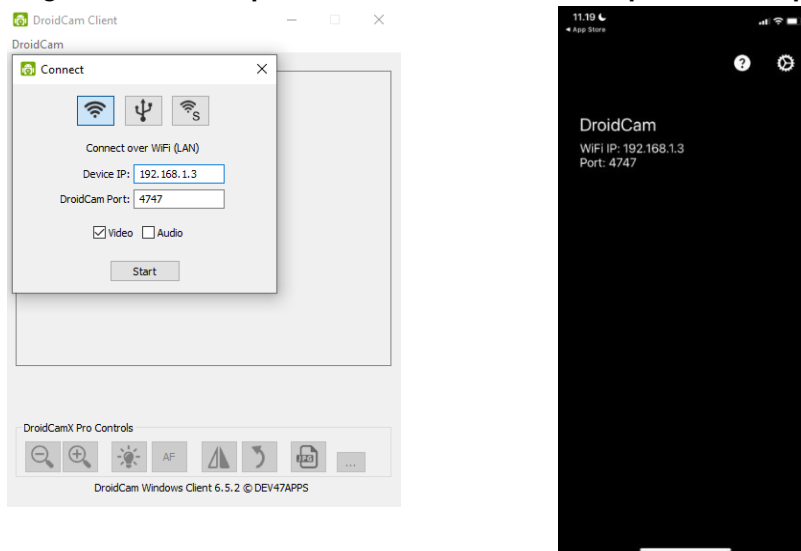
proposal	kelas	bounding box (x,y,w,h)	confidence
box 1	daun sehat	(60, 44, 148, 118)	0.92

bounding box awal disesuaikan untuk meningkatkan akurasi posisi, seperti terlihat pada output gambar akhir pada gambar 33.

hasil:

model memberikan hasil akhir berupa label kelas dan *bounding box* untuk setiap objek yang terdeteksi. gambar di atas menunjukkan hasil klasifikasi dengan label "daun sehat" dan skor kepercayaan (*confidence score*) sebesar 0.92.

3.12 konfigurasi koneksi aplikasi droidcam antara smartphone dan laptop

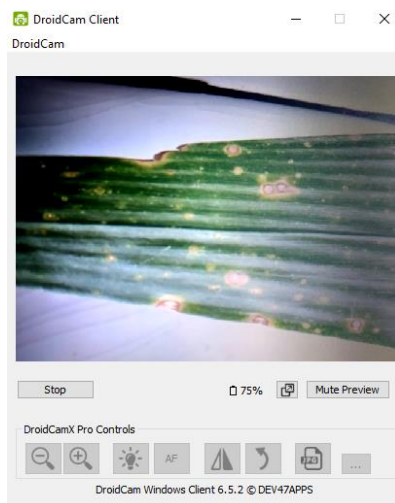


gambar 37 konfigurasi koneksi aplikasi droidcam antara smartphone dan laptop

gambar 37 menunjukkan tampilan awal aplikasi droidcam pada perangkat laptop dan smartphone yang digunakan untuk menghubungkan kamera smartphone sebagai webcam pada laptop. agar koneksi dapat berjalan dengan lancar, berikut adalah langkah-langkah yang perlu diperhatikan:

1. device ip dan wifi ip: device ip yang diinput pada aplikasi droidcam client di laptop harus sama dengan wifi ip yang ditampilkan pada aplikasi droidcam di smartphone. kedua perangkat harus berada dalam jaringan wifi yang sama agar dapat saling terhubung. misalnya, pada gambar, device ip dan wifi ip sama-sama memiliki nilai `192.168.1.3`.
2. droidcam port: droidcam port juga harus sesuai, seperti yang terlihat pada gambar, port `4747` digunakan baik pada laptop maupun smartphone. kesamaan port ini memastikan komunikasi data dapat terjadi dengan baik.
3. pengaturan video dan audio: centang opsi video untuk mengaktifkan streaming video dari kamera smartphone ke laptop. centang opsi audio jika ingin mengirimkan suara dari mikrofon smartphone ke laptop.
4. mulai koneksi: setelah memastikan device ip, wifi ip, dan droidcam port sesuai, klik tombol start pada droidcam client di laptop. jika semua langkah dilakukan dengan benar, video dari kamera smartphone akan langsung muncul di layar laptop.

dengan konfigurasi ini, smartphone dapat digunakan sebagai. proses ini penting untuk memastikan koneksi stabil dan sesuai dengan spesifikasi jaringan yang digunakan.



gambar 38 tampilan hasil koneksi aplikasi droidcam pada smartphone dan laptop

gambar di atas menunjukkan hasil koneksi antara aplikasi droidcam pada laptop dan smartphone yang telah berhasil terhubung. tampilan di laptop memperlihatkan jendela droidcam yang menampilkan video real-time dari kamera smartphone, sedangkan layar smartphone menunjukkan aplikasi droidcam dalam keadaan aktif dengan gambar yang sama. ini menegaskan bahwa kamera smartphone yang terhubung melalui droidcam berhasil digunakan sebagai webcam pada laptop untuk melakukan streaming video real-time guna mendeteksi penyakit pada tanaman jagung.