

BAB II KAJIAN PUSTAKA

2.1 Penelitian Relevan

Penelitian Relevan Sebagai pertimbangan dalam penyusunan penelitian ini digunakan beberapa sumber penelitian sebelumnya yang pernah dilakukan sebagai bahan perbandingan dan uji coba . Adapun hasil penelitian yang dijadikan perbandingan ditinjau dari segi kasus, objek penelitian, dan metode yang digunakan dalam penelitian juga tidak terlepas dari topik penelitian. Berikut merupakan beberapa penelitian yang digunakan sebagai bahan rujukan.

2.1.1 Pengenalan Jenis Kelamin Mahasiswa Universitas Nurul jadid (UNUJA) Pada Video Berdasarkan Busana Menggunakan Metode Haar Cascade dan Deep Learning

Penelitian yang dilakukan oleh Rodiatu Solehah, yang berjudul “Pengenalan Jenis Kelamin Mahasiswa Universitas Nurul Jadid (UNUJA) Pada Video Berdasarkan Busana Menggunakan Metode Haar Cascade dan Deep Learning” (Solehah, 2019). Tujuan dari penelitian ini untuk mengklasifikasi jenis kelamin mahasiswa berdasarkan busana pada area wajah secara otomatis. Metode yang digunakan yaitu Haar Cascade untuk mendeteksi dengan baik apabila wajah menghadap kamera, sedangkan metode Deep Learning digunakan untuk mengklasifikasi jenis kelamin. Pengujian gambar dilakukan dengan menggunakan 100 gambar yang terdiri dari 50 gambar laki-laki dan 50 gambar perempuan. Pengimplementasian pada gambar menghasilkan nilai akurasi 99% dan hasil uji coba pada video objek area terdeteksi dan terklasifikasi dengan baik. Dan hasil dari uji coba lebih menggunakan separuh badan sehingga tidak cocok diterapkan pada video seluruh tubuh dalam pengambilan jarak jauh. Metode Haar cascade bisa mendeteksi semua model yang sama meski itu bukan objek yang di deteksi

2.1.2 Klasifikasi Citra Multi Wajah Menggunakan Domain Adaptiv Faster Region Convolutional Neural Network

Penelitian yang dilakukan oleh Ali Akbar Lubis. dkk, yang berjudul” *Klasifikasi Citra Multi Wajah Menggunakan Domain Adaptive Faster Region Convolutional Neural Network* “ penelitian untuk mengatasi masalah pengenalan wajah yang menggunakan metode *Deep Learning*, salah satunya adalah *Faster R-CNN* dimana metode faster rcnn untuk memprediksi kecocokan pada kumpulan dataset yang dengan cara mengekstrak ciri-ciri dari beberapa perbedaan wajah dari nilai *accuracy* yang didapat sebesar 78,33%, *recall* sebesar 90.33%, *precision* sebesar 85.33% dan *accuracy* sebesar 78.33%, dan metode ini lebih ke satu *object* wajah, dan tidak cocok untuk *full body*

2.1.3 Implementasi Metode Convolutional Neural Network (CNN) Pada Pengenalan Objek Video CCTV

Penelitian yang dilakukan oleh Vandiel Mahaputra Salawazo. dkk, yang berjudul” *Implementasi Metode Convolutional Neural Network (CNN) Pada Pengenalan Objek Vidio CCTV*” dengan menggunakan metode CNN mampu mendeteksi beberapa wajah yang terdapat pada sebuah citra digital, yang merupakan citra *frame* hasil ekstrak dari video CCTV dengan tingkat akurasi sebesar 80 % untuk objek yang telah terdaftar dalam *database*, dan dapat membedakan objek yang belum terdaftar dalam *database* dengan tingkat akurasi sebesar 40 %. Dengan semakin berkembangnya teknologi maka CCTV juga mengalami perkembangan dengan pesat fungsi dari CCTV ialah untuk mengawasi lingkungan yang ada disekitar salah satunya ialah untuk mengawasi sistem keamanan. Oleh karena itu metode CNN sangat bermanfaat dalam hal ini karena metode ini dapat mengklasifikasikan objek berdasarkan ciri dari objek tersebut.

2.1.4 Deteksi Alat Pelindung Diri Menggunakan Metode Yolo Dan Faster R-CNN

Penelitian yang dilakukan oleh Jonathan Adiwibowo 1 dkk, yang berjudul” *Deteksi Alat Pelindung Diri Menggunakan Metode YOLO dan Faster R-CNN* “ Penelitian ini menggunakan dua metode, yaitu *You Only Look Once (YOLO)* dan

Faster Region-Convolutional Neural Network (Faster R-CNN). Metode YOLO digunakan untuk menemukan wilayah kepala pekerja sementara metode *F-RCNN* yang lebih cepat digunakan untuk mengklasifikasikan peralatan pelindung pribadi yang digunakan oleh pekerja. Hasil klasifikasi *R-CNN* yang lebih cepat dihitung menggunakan *confusion matrix* untuk mendapatkan akurasi prediksi yang benar. Hasil dari penelitian ini dapat mengidentifikasi pekerja yang menggunakan peralatan pelindung pribadi dalam video. Akurasi rata-rata yang telah diperoleh adalah 93,61%. Dari penelitian ini dapat disimpulkan bahwa untuk metode *Faster R-CNN* akan lebih bagus jika *dataset* yang di buat lebih banyak dan berbeda-beda model.

2.1.5 Ekstraksi Ciri untuk Klasifikasi Jenis Kelamin berbasis Wajah menggunakan Metode *Compass Local Binary Patterns*

Penelitian yang dilakukan oleh Muhammad Wahyu Budi Santoso dkk, yang berjudul "Ekstraksi Ciri untuk Klasifikasi Jenis Kelamin berbasis Citra Wajah menggunakan Metode *Compass Local Binary Patterns*". *Compass Local Binary Patterns* (CoLBP) adalah salah satu metode pengolahan citra yang digunakan sebagai ekstraksi ciri untuk klasifikasi *gender* berbasis citra wajah. CoLBP memanfaatkan *Kirsch Compass Mask* untuk meningkatkan kinerja dari *Local Binary Patterns* (LBP) dalam proses ekstraksi ciri. Pada penelitian ini menggunakan *dataset Color FERET* yang berisi foto wajah (dengan aksesoris dan tanpa aksesoris) metode klasifikasi *Random Forest* untuk proses evaluasi. Hasil penelitian ini memperoleh nilai akurasi sebesar 91,8%. Dari penelitian ini dapat disimpulkan bahwa metode CoLBP memberikan kinerja ekstraksi ciri yang baik dan aksesoris pada wajah memberikan pengaruh pada menurunnya kualitas ciri yang dihasilkan metode CoLBP. Kelemahan dari penelitian ini hanya bisa untuk wajah saja tidak untuk *full body*.

Berdasarkan penelitian diatas maka terdapat perbedaan dengan penelitian yang akan dilakukan saat ini. Penelitian sebelumnya menggunakan metode Deteksi Tepi dan *Backpropagation*, metode *Convolution neural network* (CNN), metode *Haar Cascade Classifier* dan objek yang digunakan pada penelitian sebelumnya hanya wajah saja. Sedangkan pada penelitian ini menggunakan metode *Faster Region Convolutional neural Network (Faster R-CNN)*, algoritma terbaru berbasis wilayah

terbaru yang menunjukkan hasil luar biasa pada berbagai deteksi objek seluruh tubuh.

2.2 Landasan teori

2.2.1 Jenis kelamin

Jenis kelamin atau *gender* merupakan penggolongan secara gramatika dari perbedaan antara perempuan dengan laki-laki secara biologis adapun laki-laki sudah terlihat jelas dari postur tubuhnya kuat dan produktif sedangkan perempuan secara biologis terlihat lembut, dari arah jalannya dan bertutur kata. (2016:43).

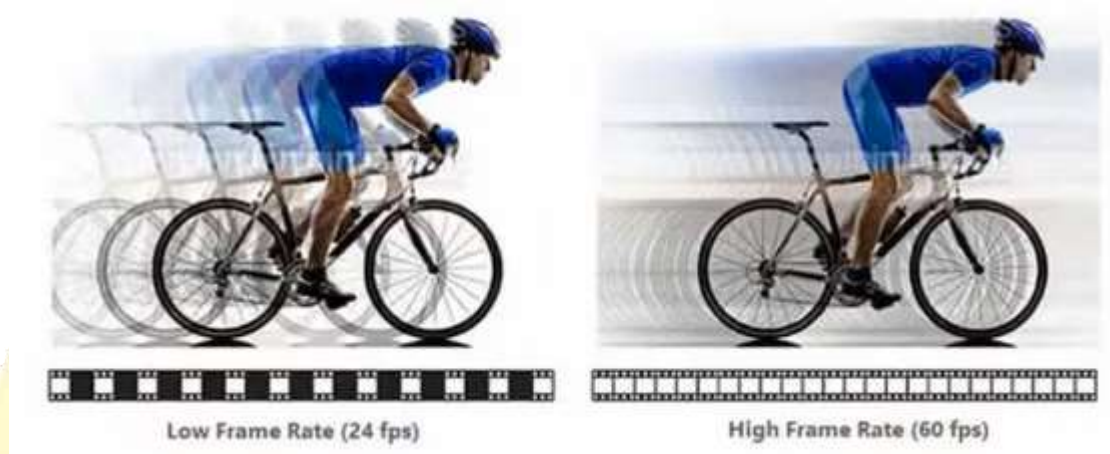
2.2.2 Busana

Busana pada umumnya suatu ekspresi atau ungkapan pribadi yang tidak selalu sama untuk setiap orang. Perubahan mode yang menyangkut busana akan terjadi lebih cepat dibandingkan dengan perubahan kebudayaan secara keseluruhan. Dalam dunia *fashion* terdapat istilah *fashionable* dan *unfashionable* untuk menjelaskan apakah seseorang tersebut mengikuti perkembangan mode terbaru atau tidak. Adanya ruang lingkup pesantren busana muslim sangatlah indetik bagi kalangan santri, busana muslim yang biasa dipakai di pesantren ialah sarung, kopyah, gamis, dan Rok yang sangat rapi dan memukau.

2.2.3 Video Digital atau Video Analisis

Video pada dasarnya merupakan *array* tiga dimensi yang terdiri dari dua dimensi untuk menggambarkan ruang pergerakan citra (*spasial*) dan satu dimensi lainnya menggambarkan waktu. Video digital tersusun atas serangkaian *frame* yang ditampilkan dengan kecepatan tertentu (*frame/detik*) (Priutomo, Magdalena, & Andini, 2016). Mata manusia melihat sebuah video digital sebagai rangkaian yang kontinyu dikarenakan laju *frame* yang tinggi. *Frame* adalah representasi dari suatu gambar atau citra digital. Suatu citra digital direpresentasikan dengan sebuah matriks yang masing-masing elemennya merepresentasikan nilai intensitas atau kedalaman warna. Karakteristik yang dimiliki oleh sebuah video digital menentukan kualitas dari video tersebut. Hal ini akan berpengaruh terhadap sensitivitas mata manusia dalam melihat video digital. Video digital dapat dikatakan baik dilihat dari karakteristiknya,

adapaun karakteristik yang dimaksud diantaranya adalah Resolusi, kedalaman bit dan laju *frame*. Video juga merupakan serangkaian gambar statis dalam urutan tertentu sehingga gambar statis bisa disebut dengan *frame* dari *frame* tersebut akan terstruktur menjadi video bergerak (Priutomo, Magdalena, & Andini, 2016)



Gambar 2.1 Penerapan Video Analisis

2.2.4 CCTV (Closed Circuit Television)

CCTV atau *Closer Circuit Television* (CCTV) pertama kali ditemukan oleh Walter Brunch dan digunakan oleh tim pelaksana untuk peluncuran roket V-2 yang dilakukn pada tahun 1942 di Jerman.. Adapun Fungsi kamera CCTV adalah untuk memantau keadaan dalam suatu tempat, yang biasanya berkaitan dengan keamanan(*seccurity*). Umumnya kamera CCTV dipasang pada tempat-tempat umum seperti bank, bandara, hotel, tempat atm, dll. Pada saat-saat tertentu kamera CCTV akan sangat berguna sebagai barang bukti, seperti ketika terjadi bencana besar atau peristiwa-peristiwa penting yang tidak sempat dipantau oleh manusia. Berdasarkan pengertian CCTV di atas, maka fungsi dari CCTV sangat penting untuk

membantu pekerjaan manusia dalam menjaga keamanan. Di dalam Pondok Pesantren CCTV lebih umum digunakan ketika santri yang keluar masuk pondok, dan pertemuan santri putra dan santri putri yang dilarang oleh pihak pesantren dari hasil CCTV yang sudah terekam maka disitulah keamanan pesantren bisa memantaunya.



Gambar 2.2 Beberapa bentuk CCTV

2.2.5 Google Colaboratory

Google Colaboratory atau Google Colab merupakan sebuah *tools* baru yang dikeluarkan oleh Google. *Tools* ini juga memberikan fasilitas pada pengguna atau peneliti yang ingin mempelajari *machine learning* maupun *Deep Learning*, namun *tools* ini masih memiliki keterbatasan perangkat dalam melakukan komputasi. Selain itu Google Colab juga menyediakan layanan GPU gratis pada pengguna sebagai *backend* komputasi yang digunakan selama 12 jam. Google Colab dibuat diatas Environment Jupyter sehingga mirip seperti Jupyter Notebook. Untuk penggunaannya sama seperti dengan Jupyter Notebook yang membedakannya hanya dalam media penyimpanannya. Media penyimpanan pada Google Colab adalah Google Drive dan *tools* yang berjalan pada sistem *Cloud*. Dengan menggunakan Google Colab maka dapat digunakan pada aplikasi berbasis *Deep Learning* dengan *library* seperti keras, *tensorflow*, *python* dan *opencv*. (Digmi, 2018)

2.2.6 Machine Learning

Istilah *Machine Learning* pertama kali didefinisikan oleh Arthur Samuel pada 1959, menurut Arthur Samuel, *machine learning* adalah salah satu bidang ilmu komputer yang memberikan kemampuan pembelajaran kepada komputer untuk

mengetahui sesuatu tanpa pemrograman yang jelas. *Machine Learning* dapat didefinisikan sebagai metode komputasi berdasarkan pengalaman untuk meningkatkan performa atau membuat aplikasi yang akurat. Pengalaman yang dimaksud yaitu informasi yang sebelumnya telah tersedia dan bisa dijadikan data pembelajaran. Fokus dari penelitian *Machine Learning* adalah bagaimana mengenali secara otomatis pola kompleks dan membuat keputusan cerdas berdasarkan data. Dalam pembelajaran *machine learning*, terdapat beberapa skenario.

a. *Supervised Learning*

Penggunaan skenario *supervised learning*, pembelajaran menggunakan masukan dari beberapa data pembelajaran yang telah diberi label atau id. Setelah itu membuat prediksi dari data yang telah diberi label. Contoh: *k-Nearest Neighbors*, *Naive Bayes*, *Support vector machines*, *Decision trees*.

b. *Unsupervised Learning*

Penggunaan skenario *Unsupervised Learning*, pembelajaran menggunakan masukan data pembelajaran yang tidak diberi label. Setelah itu mencoba untuk mengelompokan data berdasarkan karakteristik-karakteristik yang ditemui. Contoh: *k-Means*, *DBSCAN*.

c. *Reinforcement Learning*

Pada skenario *reinforcement learning* fase pembelajaran dan tes saling dicampur. Untuk mengumpulkan informasi pembelajar secara aktif dengan berinteraksi di lingkungan sehingga untuk mendapatkan balasan untuk setiap aksi dari pembelajar.

Saat ini telah banyak pendekatan *machine learning* yang digunakan untuk deteksi spam, *Optical Character Recognition* (OCR), pengenalan wajah, deteksi penipuan online, *NER* (*Named Entity Recognition*), *Part-of-Speech Tagger* (Solehah, 2019).

d. *Deep Learning*

Deep Learning merupakan salah satu bidang dari *machine learning* yang memanfaatkan jaringan syaraf tiruan untuk implementasi permasalahan dataset dengan jumlah yang besar. Dengan memanfaatkan banyak layer atau sering disebut

Multi Layer Perceptron (MLP) dalam pengolahan informasi non-linier untuk melakukan ekstraksi fitur, klasifikasi, dan pengolahan data. Pada metode *machine learning* terdapat teknik untuk mengekstraksi fitur dari data pelatihan dan algoritma pembelajaran khusus untuk mengklasifikasi citra maupun untuk mengenali suara. Namun, metode *machine learning* memiliki kekurangan dalam kecepatan dan akurasi (Nurhikmat, 2018).

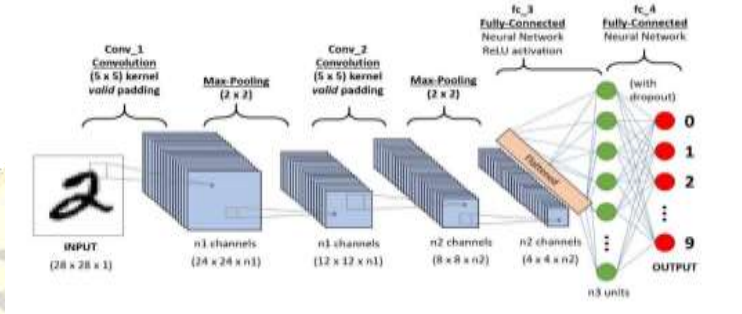
Pendekatan dalam menyelesaikan masalah *Deep Learning* menggunakan konsep hierarki. Konsep yang dapat membuat komputer mempelajari konsep yang kompleks dengan menggabungkan konsep-konsep yang sederhana. Jadi dapat digambarkan sebuah graf bagaimana konsep tersebut dibangun di atas konsep yang lain, graf ini akan dalam banyak layer, dan hal ini alasan disebut sebagai *Deep Learning* (Dewi, 2018).

e. ***Convolutional Neural Network (CNN)***

Convolutional Neural Network (CNN) adalah algoritma jaringan saraf tiruan perkembangan dari Deep Neural Network sehingga termasuk keluarga Neural Network dan merupakan algoritma supervised learning. Algoritma ini paling umum diterapkan untuk menganalisis citra visual. Dalam penggunaan aplikasinya dalam pengenalan gambar dan video, sistem pemberi rekomendasi [10], klasifikasi gambar, analisis gambar medis, pemrosesan bahasa alami, dan rangkaian waktu keuangan.

Convolutional Neural Network (CNN) pertama kali dikembangkan dengan nama NeoCognitron oleh Kuniyiko Fukushima, seorang peneliti dari NHK Broadcasting Science Research Laboratories, Kinuta, Setagaya, Tokyo, Jepang [4]. Konsep tersebut kemudian dimatangkan oleh Yann LeCun, seorang peneliti dari AT&T Bell Laboratories di Holmdel, New Jersey, USA. Model CNN dengan nama LeNet berhasil diterapkan oleh LeCun pada penelitiannya mengenai pengenalan angka dan tulisan tangan [1]. Pada tahun 2012, Alex Krizhevsky dengan penerapan CNN miliknya berhasil menjuarai kompetisi ImageNet Large Scale Visual Recognition Challenge 2012. Prestasi tersebut menjadi momen pembuktian bahwa metode *Deep Learning*, khususnya CNN. Metode CNN terbukti berhasil

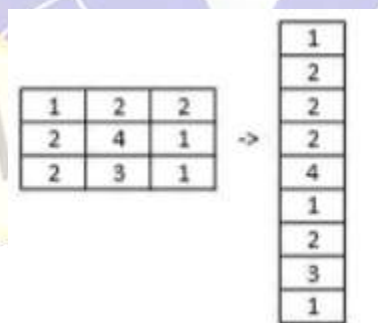
mengungguli metode Machine Learning lainnya seperti SVM pada kasus klasifikasi objek pada citra.



Gambar 2.3. Struktur Convolutional Neural Network

Arsitektur CNN dianalogikan seperti pola konektivitas *neuron* di otak manusia dan terinspirasi oleh organisasi di korteks visual manusia. Neuron secara individu merespon rangsangan hanya di daerah terbatas bidang visual yang dikenal sebagai bidang reseptif. Kumpulan bidang tersebut tumpang tindih untuk menutupi seluruh area visual.

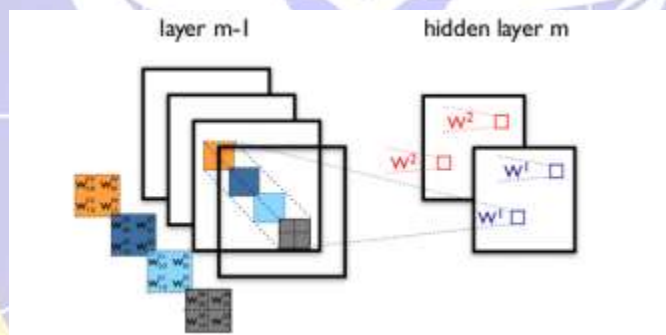
Untuk melakukan klasifikasi menggunakan CNN khususnya pada gambar, gambar harus diubah terlebih dahulu menjadi angka. Gambar adalah sebuah nilai *pixel* dalam matriks, sehingga jika matriks gambar 3x3 maka bisa diubah menjadi matriks vektor 9x1 dan memasukkannya ke *Multi-Level Perceptron* untuk keperluan klasifikasi.



Gambar 2.4 Pengubahan Matriks 3x3 ke Matriks 9x1

Multi-Level Perceptron diatas hanya akan bekerja dalam gambar biner yang sangat dasar, dimana metode ini Mungkin menunjukkan skor presisi rata-rata saat melakukan klasifikasi kelas tetapi akan sedikit atau bahkan tidak bisa ketika melakukan klasifikasi gambar yang kompleks yang tidak memiliki ketergantungan piksel di seluruhnya.

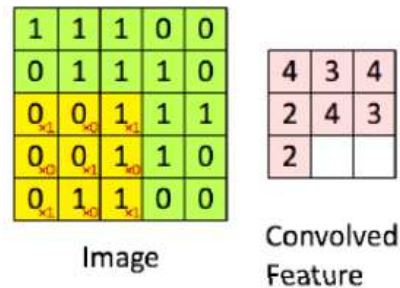
Dengan menggunakan CNN, *dependensi spasial* dan *temporal* pada gambar kompleks dapat ditangkap oleh algoritma melalui penerapan filter yang relevan. Pada Arsitektur CNN melakukan pemasangan dan penerapan filter yang lebih baik dengan *dataset* gambar untuk pengurangan jumlah parameter dan usabilitas bobot. Dengan kata lain, jaringan syaraf dapat dilatih untuk memahami gambar dengan lebih baik. Konsep dan Cara kerja CNN memiliki kesamaan pada MLP, namun dalam CNN setiap *neuron* dipresentasikan dalam bentuk dua dimensi, tidak seperti MLP yang setiap *neuron* hanya berukuran satu dimensi



Gambar 2.5 Proses Konvolusi pada CNN

Convolution Layer melakukan operasi konvolusi pada *output* dari *layer* sebelumnya. *Layer* tersebut adalah proses utama yang mendasari sebuah CNN. Konvolusi adalah suatu istilah matematis yang berarti mengaplikasikan sebuah fungsi pada *output* fungsi lain secara berulang. Dalam pengolahan citra, konvolusi berarti mengaplikasikan sebuah *kernel* (kotak kuning) pada citra disemua *offset* yang memungkinkan seperti yang ditunjukkan pada Gambar 2.4. Kotak hijau secara keseluruhan adalah citra yang akan dikonvolusi. *Kernel* bergerak dari sudut kiri atas ke kanan bawah. Sehingga hasil konvolusi dari citra tersebut dapat dilihat pada gambar disebelah kanannya. Tujuan dilakukannya konvolusi pada data citra adalah untuk mengekstraksi fitur dari citra input. Konvolusi akan menghasilkan transformasi

linear dari data input sesuai informasi spasial pada data. Bobot pada *layer* tersebut menspesifikasikan *kernel* konvolusi yang digunakan, sehingga *kernel* konvolusi dapat dilatih berdasarkan input pada CNN.



Gambar 2.6 Operasi Konvolusi

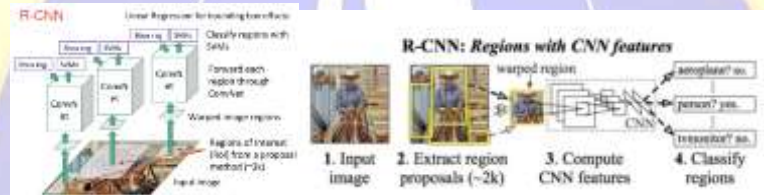
f. Region-Convolutional Neural Network

Region Convolutional Neural Network adalah algoritma jaringan saraf tiruan dan perkembangan lanjutan dari algoritma *Convolutional Neural Network*. Dimana untuk menghindari masalah memilih sejumlah besar daerah Untuk mengatasi masalah pemilihan sejumlah besar wilayah, Maka muncul lah teknik [Regions with CNN features atau R-CNN](#) yang diperkenalkan oleh peneliti dari UC Berkeley, [Ross Girshick](#) dkk, pada tahun 2014. Mengusulkan metode di mana kami menggunakan pencarian selektif untuk mengekstrak hanya 2000 wilayah dari gambar dan dia menyebutnya proposal wilayah. Oleh karena itu, daripada mencoba mengklasifikasikan sejumlah besar daerah, Anda dapat bekerja dengan 2000 daerah. 2000 proposal wilayah ini dibuat menggunakan algoritma pencarian selektif yang tertulis yaitu (1).Generate initial sub-segmentation,we generate many candidate regions, (2).Use greedy algorithm to recursively combine similar regions into larger ones .(3) Use the generated regions to produce the final candidate region proposals.

Dua ribu kandidat *Region Proposal* ini dibengkokkan menjadi kotak dan dimasukkan ke dalam jaringan saraf *konvolusional* yang menghasilkan vektor fitur 4096-dimensi sebagai *output*. CNN berfungsi sebagai *ekstraktor fitur* dan lapisan padat *output* terdiri dari *fitur* yang diekstraksi dari gambar. *Fitur* yang telah diekstraksi dimasukkan ke dalam *Support Vector Machine (SVM)* untuk mengklasifikasikan keberadaan objek dalam proposal wilayah kandidat. Algoritma

dapat memprediksi empat nilai yang *offset* untuk meningkatkan ketetapan kotak pembatas seperti suatu wilayah proposal, algoritma akan memprediksi keberadaan orang. Setelah itu dari hasil gambar orang akan di potong menjadi dua, dan dari itu tergantung *offset* yang menyesuaikan kotak pembatas proposal wilayah. Adapun kekurangan dari metode RCNN (1) Masih

membutuhkan banyak waktu untuk melatih jaringan karena harus mengklasifikasikan 2000 proposal wilayah per gambar, (2) tidak dapat diterapkan secara *real time* karena membutuhkan sekitar 47 detik untuk setiap gambar uji (3). Algoritma pencarian selektif adalah algoritma tetap. Oleh karena itu, tidak ada pembelajaran yang terjadi pada tahap itu. Gambar 2.7 Menunjukkan arsitektur R-CNN



Gambar 2.7 Proses System RCNN

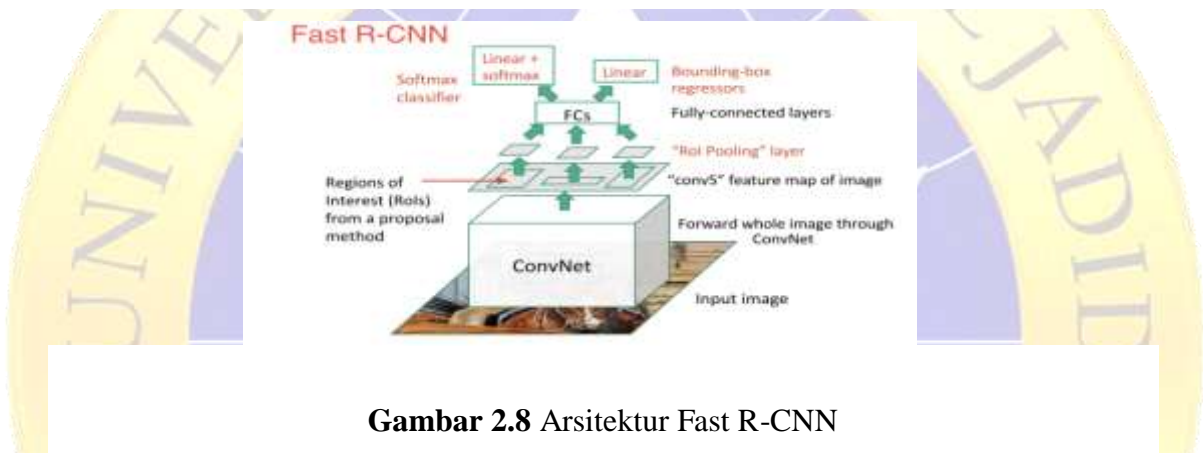
g. Fast R-CNN.

Meskipun peningkatannya cukup signifikan, proses *training* dari R-CNN memiliki banyak kendala.terutama dalam menghasilkan *region* sebelum memulai proses *training*. Selain itu, melatih CNN itu harus sendiri, kita juga perlu melakukan *training* untuk SVM yang digunakan, Oleh karena itu setahun berikutnya, yaitu pada tahun 2015, Ross Girshick mengembangkan R-CNN menjadi *Fast R-CNN*.

Adapun alur kerja dari Fast R-CNN sedikit berbeda dari sebelumnya, dimana bila sebelumnya setiap *region-region* dari *region proposal method* (Selective Search) akan memiliki CNN untuk *feature extraction* masing-masing, *Fast R-CNN* hanya menggunakan 1 CNN saja. Dari satu CNN yang dihasilkan kemudian “dicocokkan” dengan *Region of Interest* (ROI) yang didapat dari *Selective Search* tadi, untuk

kemudian diklasifikasi kelas nya dan dideteksi *bounding box* nya. Dengan kata lain, *Fast R-CNN* melakukan *feature extraction* sebelum mengajukan *region*.

R-CNN juga meninggalkan penggunaan SVM sebagai *classifiers*, menggantinya dengan ROI *pooling* dan *fully-connected layers*. Dengan pedekatan 1 CNN, ROI *pooling layer* dan *feed forward network* tidak hanya mempercepat performa R-CNN , namun dapat menambah kapabilitas R-CNN menjadi *end-to-end differentiable* dan juga mempermudah proses *training* (karena tidak perlu lagi melakukan *train* untuk SVMs dan hanya melatih 1 arsitektur CNN ketimbang banyak CNN seperti sebelumnya).



Gambar 2.8 Arsitektur Fast R-CNN

h. *Faster Region Convolutional Neural Network*

Faster R-CNN yaitu pengembangan dari *Fast R-CNN*. Sebagai terbaru dari generasi sebelumnya dari metode *object detection* menunjukkan hasil yang mengesankan pada objek deteksi *Faster Region based Convolutional Neural Network* atau biasa disebut *Faster R-CNN*. *Faster R-CNN* merupakan sebuah metode perubahan dari *Fast R-CNN*. Perubahan ini dilakukan karena terjadi *bottleneck* terhadap arsitektur pada bagian *selective search*, karena *selective search* diperlukan untuk menghasilkan 2000 proposal per gambar. Bagian tersebut merupakan bagian yang paling banyak memakai waktu dalam pelatihan. Di dalam *Faster R-CNN* *selective search* diubah menggunakan RPN (*Region Proposal Network*, RPN berfungsi untuk menghasilkan beberapa *bounding box* dimana setiap *box* memiliki 2 skor probabilitas apakah pada lokasi tersebut terdapat objek atau tidak. Dengan

adanya RPN tidak dilakukan pemrosesan berulang ulang seperti dilakukan kepada *R-CNN* dan membuat model keseluruhan dapat di *training* secara *end-to-end*.

Adapun cara kerja *Faster R-CNN* sebagai berikut:

- a) *Convolution layer* yaitu membuat lalu mengirimkan feature map kepada RPN.
- b) *RPN* yaitu feature map yang di proses lalu membuat region proposal lalu membuat bounding box kebagian yang dianggap memiliki kemungkinan terdapat objek.
- c) *R-CNN* melakukan klasifikasi terhadap proposal yang sudah dibuat RPN lalu menentukan apakah ada objek proposal yang merupakan suatu objek yang ada pada bagian model yang sudah dilatih dan membuat pelabelan pada objek

Faster R-CNN memiliki kelebihan diantaranya adalah sebagai berikut:

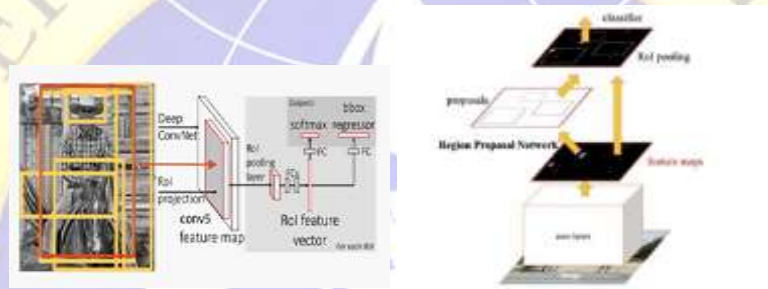
- a) Proses *training* data yang terlihat cepat karena proposal region yang sudah dibuat tidak sebesar pada *R-CNN* dimana dalam membuat proses pengerjaan lebih ringan.
- b) Dapat diimplementasikan dalam klasifikasi real time karena hanya membutuhkan pemrosesan waktu yang relatif singkat yaitu sekitar 1 – 2 detik

Penjelasan arsitektur umum dari *Faster R-CNN*:

- a) *Convolutional Layer* : Pada tahap ini convolutional layer dapat mempelajari dalam bagian penting yang menjadi ciri khas objek tersebut serta membuat feature map dari objek yang sudah ditangkap kamera CCTV, Convolutional adalah layer yang terdiri dari neuron yang sudah tersusun sedemikian rupa sehingga membentuk sebuah dua dimensi dan filter dengan panjang dan tinggi (pixels).
- b) *Feature Map* : *Feature Map* adalah sebuah map yang dibuat dari *convolutional layer* yang berisi informasi tentang *representasi* vector dari gambar atau frame yang ditangkap.
- c) *RPN (Region Proposal Network)* : *RPN* merupakan sebuah modul yang bekerja dalam mengolah feature map yang telah dibuat pada *convolution layer* untuk memprediksi bagian yang memungkinkan sebagai objek dan melakukan prediksi bounding box dari objek tersebut, RPN dibagi menjadi 2 *convolution*

layer dimana 1 layer bertanggung jawab untuk mendeteksi letak objek dan 1 layer berfungsi memprediksi bounding box.

- d) *ROI Pooling* : *ROI* merupakan layer yang bertugas untuk mengekstrak informasi *feature map* yang sudah diproses oleh RPN untuk dikasifikasi pada *classification layer*.
- e) *Classification Layer* : *Classification layer* merupakan layer yang berfungsi dalam mengelompokkan objek yang telah dideteksi pada RPN dan melakukan pelabelan terhadap objek tersebut serta memberikan bounding box pada objek tersebut.



Gambar 2.9 Arsitektur dari *Faster R-CNN* (Le, 2020)

Gambar 2.10 menunjukkan perbedaan arsitektur dari metode yang telah disebutkan sebelumnya.

	R-CNN	Fast R-CNN	Faster R-CNN
Test time per image	50 seconds	2 seconds	0.2 seconds
Speed-up	1x	25x	250x
mAP (VOC 2007)	66.0%	66.9%	66.9%

Gambar 2.10 Perbedaan Arsitektur *R-CNN*, *Fast R-CNN*, dan *Faster R-CNN*

2.2.7 *You Only Look Once (YOLO)*

You Only Look Once (YOLO) adalah sebuah algoritma yang dikembangkan untuk mendeteksi sebuah objek secara *real-time*. pendeteksian system yang dilakukan dengan menggunakan *repurpose classifier* atau *localizer* untuk melakukan deteksi.

adapun model diterapkan pada sebuah citra dari beberapa lokasi dan skala. Daerah dengan citra yang diberi *score* sangat tinggi akan dianggap sebagai sebuah pendeteksian (Jupiyandi, Saniputra, Pratama, Dharmawan, & Cholissodin, 2019). YOLO menggunakan pendekatan jaringan syaraf tiruan (JST) dalam mendeteksi sebuah objek pada citra. Jaringan ini membagi citra menjadi beberapa wilayah dari prediksi setiap kotak probabilitas dan pembatas dalam setiap wilayah. Kotak-kotak pembatas ini lalu dibandingkan dengan setiap probabilitas yang diprediksi (Jupiyandi, Saniputra, Pratama, Dharmawan, & Cholissodin, 2019). YOLO memiliki beberapa kelebihan dibandingkan dengan sistem yang berorientasi pada *classifier*, terlihat dari seluruh citra pada saat tes yang dilakukan dalam prediksi yang diinformasikan secara global pada citra (Redmon, Divvala, Girshick, & Farhadi, 2016). Hal tersebut juga membuat prediksi dengan sintesis jaringan saraf ini tidak seperti sistem *Region-Convolutional Neural Network* (RCNN) yang membutuhkan sebuah ribuan *dataset* dalam sebuah citra sehingga YOLO membuat lebih cepat beberapa kali dari pada R-CNN (Jupiyandi, Saniputra, Pratama, Dharmawan, & Cholissodin, 2019). Gambar 2.10 adalah contoh hasil deteksi objek dengan menggunakan algoritma YOLO 2.7



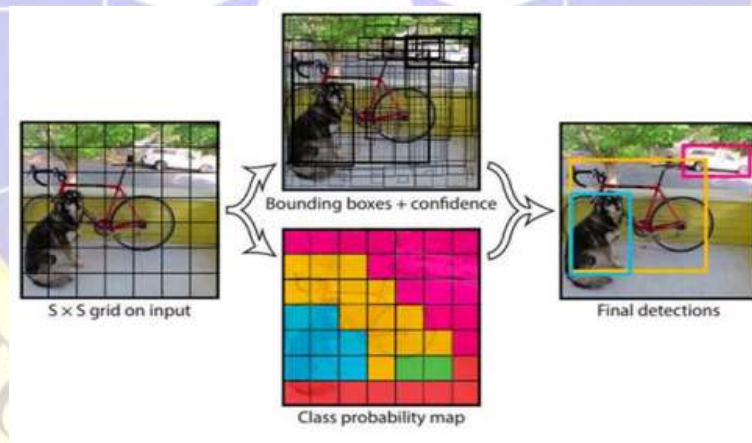
Gambar 2.11 Deteksi objek dengan YOLO

YOLO mendeteksi objek dalam beberapa tahapan, diantaranya adalah (Asshiddiqie, Rahmat, & Anggraeny, 2020):

1. Membagi citra dalam *region/grid* berukuran $s \times s$. *Grid* tersebut bertanggung jawab dalam mendeteksi objek. Pada gambar 2.1.1, tiap *grid* akan diprediksi oleh *bounding box* beserta nilai *confidence*. Nilai *confidence* menunjukkan

seberapa yakin *bounding box* tersebut berisi objek dan seberapa akurat prediksinya.

2. Tiap *bounding box* memiliki 5 nilai informasi yaitu x , y , w , h , dan c . Nilai x dan y adalah koordinat titik tengah *bounding box* yang terprediksi, nilai w dan h adalah rasio ukuran lebar dan tinggi relatif terhadap *grid*, dan c adalah nilai *confidence bounding box* tersebut.
3. Pada algoritma YOLO, setiap *grid* akan memprediksi nilai kelas probabilitas jika diprediksi terdapat objek di dalamnya saat pengujian, maka YOLO akan mengalikan 80 nilai *class probability* dengan nilai *confidence* dari *bounding box*. Sehingga dapat menghasilkan nilai *confidence* kelas secara spesifik pada tiap *bounding box*. Nilai tersebut menunjukkan *class probability* yang muncul pada *bounding box* dan seberapa akurat *bounding box* dalam memprediksi sesuai dengan objek (Redmon, Divvala, Girshick, & Farhadi, 2016). Gambar 2.12. Menunjukkan gambaran diagram YOLO.



Gambar 2.12. Diagram algoritma YOLO.

