

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Dalam beberapa tahun terakhir, penelitian mengenai *Object Detections* untuk mendeteksi kendaraan telah dilakukan. Mulai dari menggunakan *Image Based Detection* ataupun *Sound Based Detection*. Perbedaan antara *Image Based* atau *Sound Detection* adalah bentuk input nya. Sesuai dengan namanya *Image Based Detection* adalah mendeteksi objek dari suatu gambar. *Sound Based Detection* adalah deteksi objek berdasarkan suatu file suara. Pada penelitian ini metode yang di pakai adalah *Image Based Detection* karena pada *Sound Based Detection* berpotensi terjadi nya *False Alarm* cukup besar yang disebabkan oleh suara bising dari kendaraan lain ataupun suara pekerja bangunan atau jalan. Dan juga *Sound Detection System* dapat terganggu oleh kondisi cuaca, contoh nya ketika hujan lebat dan angin yang kencang.

Penulis juga mengkaji penelitian terdahulu bertujuan agar mendapatkan bahan kajian untuk perbandingan dan acuan. Selain itu, untuk menghindari anggapan plagiarism dengan penelitian yang penulis buat ini. Maka dari itu peneliti mencantumkan hasil- hasil penelitian terdahulu sebagai berikut :

1. Penelitian yang di lakukan oleh Bilel Benjdira, Taha Khursheed, AnisKouba, Adel Ammar dan Kais Ouni.

Penelitian yang berjudul “Car Detection using Unmanned Aerial Vehicles: Comparison between Faster R-CNN and YOLOv3”. Penelitian ini pada intinya adalah untuk mendeteksi kendaraan mobil lewat pesawat tanpa awak dan bertujuan untuk membandingkan algoritma *computer vision* yang sering di pakai untuk object detection. Algoritma yang dibandingkan adalah FasterR-CNN dan YOLOv3. Berdasarkan penelitian tersebut,

disimpulkan bahwa YOLOv3 mengungguli R-CNN dalam segi kecepatan pemrosesan dan juga lebih sensitif dalam mendeteksi kendaraan mobil. Nilai sensitivitas dari YOLOv3 dalam penelitian ini adalah 99.07% sedangkan Fast R-CNN adalah 79.40%.

Sedangkan waktu untuk mendeteksi dan memprosesnya YOLOv3 memakan waktu 0.057 ms dan Fast R-CNN 1.39 s. Maka dari itu penulis memilih algoritma YOLOv3 berdasarkan hasil penelitian jurnal tersebut.

2. Penelitian oleh Mawaddah Harahap , Juni Elfrida , Pasrah Agusman , Mario Rafael , Rahul Abram , Kiki Andrianto.

Penelitian yang berjudul “Sistem Cerdas Pemantauan Arus Lalu Lintas Dengan YOLO (You Only Look Once v3)” merupakan jurnal yang peneliti gunakan selanjutnya. Penelitian ini bertujuan untuk implementasi algoritma YOLOv3 untuk memantau arus lalu lintas dengan media CCTV di kota Medan. Nantinya sistem ini dapat mengidentifikasi jenis kendaraan lewat CCTV di kota tersebut yang terletak di 9 persimpangan jalan. Peneliti jurnal ini mengungkapkan alasan menggunakan YOLOv3, yaitu karena algoritma tersebut merupakan salah satu metode yang cepat dan akurat. Dari penelitian ini nilai mAP dari setiap persimpangan menghasilkan nilai di atas 90%.

U M N
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

3. Penelitian oleh Nur Fadlia, Rifki Kosasih

Penelitian yang berjudul “Klasifikasi Jenis Kendaraan Menggunakan Metode CNN” merupakan jurnal yang peneliti gunakan selanjutnya. Penelitian ini bertujuan untuk klasifikasi jenis-jenis kendaraan yang berjalan dan tidak sesuai pada jalurnya untuk mengurangi angka kemacetan pada wilayah depok. Dataset terdiri dari 120 dataset citra yang berisi citra mobil, motor, dan sepeda. Hasil menunjukkan akurasi 17 sebesar 94,4% pada tahap pelatihan dan 73,3% pada tahap pengujian.

4. Penelitian oleh Joseph Redmon, AliFarhadi

Penelitian ini berjudul “Yolo V3 :an Incremental Improvement” merupakan jurnal lengkap yang membahas dan mem breakdown algoritma Yolo mulai dari training, testing hingga evaluation. Pada jurnal ini juga dijabarkan perbandingan YOLOv3 dengan Fast-RCNN. Hasil dari penelitian tersebut adalah YOLOv3 merupakan algoritma Object Detection yang cocok di gunakan untuk sistem yang Real time karena menghasilkan akurasi yang bagus serta nilai mAP yang baik dan high speed detector.

5. Penelitian oleh Muhammad Usaid, Muhammad Asif, Tabarka Rajab, Munaf Rashid dan Syda Iqra Husain

Penelitian berjudul “Ambulance Siren Detection using Artificial Intelligence in Urban Scenarios” merupakan jurnal yang membahas tentang perancangan sistem mendeteksi ambulans dengan menggunakan sirine ambulans. Tujuan penelitian ini adalah untuk mendeteksi suara sirene ambulans agar pengendara jalan dapat mendengar sirine tersebut kemudian memberikan jalan kepada ambulans sehingga ambulans dapat sampai tujuan dengan tepat waktu. Hasil penelitian ini adalah dihasilkan akurasi sebesar 90%.

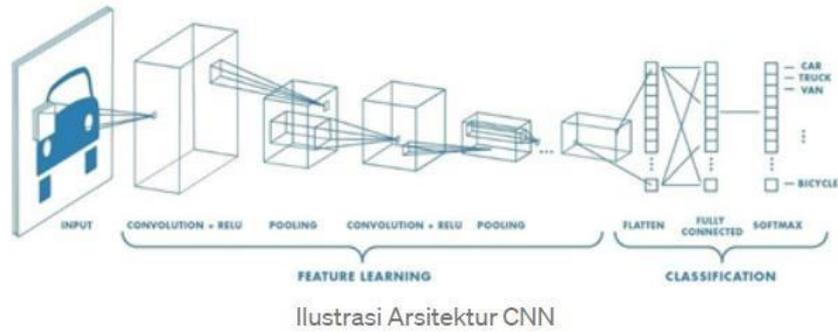
6. Penelitian oleh Zubayer Islam dan Mohamed Abdel

Penelitian yang berjudul “Realtime Emergency Vehicle Event Detection Using Audio Data” merupakan jurnal yang bertujuan untuk mendeteksi kendaraan darurat menggunakan audiodata dan membandingkan berdasarkan Run Time dari beberapa algoritma. Metode yang di gunakan adalah Extreme Learning Machine. Hasil dari penelitian tersebut adalah CNN mempunyai Run time yang lama dan menghasilkan akurasi yang rendah jika menggunakan *audio based detection*. Maka dari itu penulis memilih *image based detection*.

2.2 Tinjauan Teori

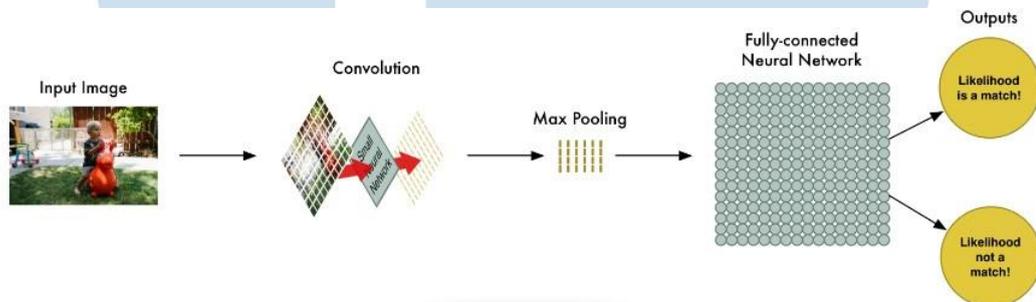
2.2.1 Convolutional Neural Network (CNN)

Convolution Neural Network merupakan salah satu jenis metode dari *Deep Neural Network* yang di dalam nya terdapat tingkatan jaringan yang berfungsi untuk mengekstrak fitur – fitur dari sebuah citra. Secara garis besar, CNN memanfaatkan proses konvolusi dengan menggerakkan sebuah kernel konvolusi (filter) berukuran tertentu ke sebuah gambar, komputer mendapatkan informasi representatif baru dari hasil perkalian bagian gambar tersebut dengan filter yang digunakan. Struktur CNN di bagi menjadi 2 bagian besar, yaitu *Feature Extraction Layer* dan *Fully Connected Layer*.



Gambar 2.1 Struktur CNN

Sumber : (<https://medium.com/@16611110/apa-itu-convolutional-neural-network-836f70b193a4>)



Gambar 2.2 Struktur CNN

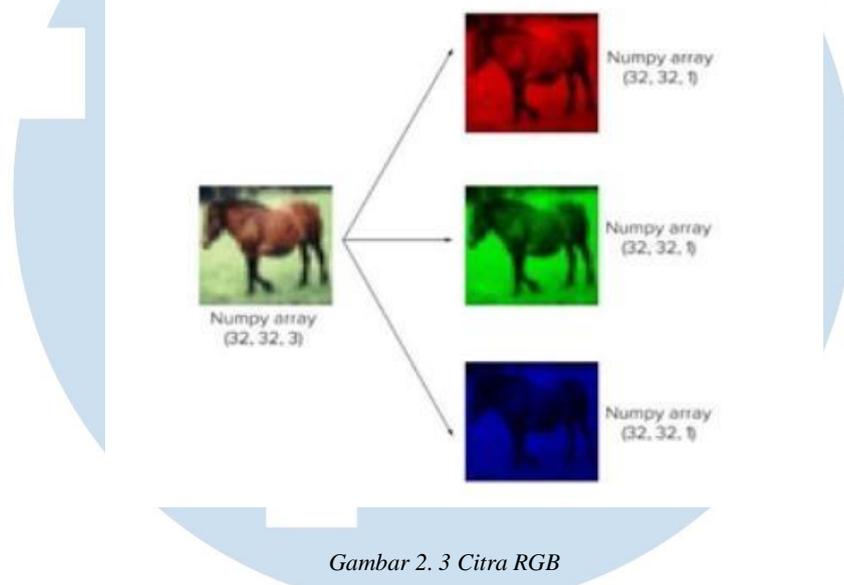
Sumber : (<https://medium.com/@16611110/apa-itu-convolutional-neural-network-836f70b193a4>)

Pada *Feature Extraction Layer* terjadi proses *encoding* dari yang awalnya *image* menjadi *features* berupa angka – angka yang mempresentasikan *image* tersebut (*Feature Extraction*). *Feature Extraction* sendiri dibagi menjadi dua bagian, yaitu *Convolutional Layer* dan *Pooling Layer*. Namun jika dijabarkan secara lengkap komponen – komponen dari CNN adalah :

- *Input Layer*
- *Convolution Layer*
- *Activation Function*
- *Pooling Layer*
- *Fully Connected Layer*

A. *Input Layer*

Input Layer merupakan lapisan awal dari struktur CNN yang terdiri dari *artificial neuron* dan membawa data ke dalam sistem untuk di proses lebih lanjut oleh lapisan berikutnya.



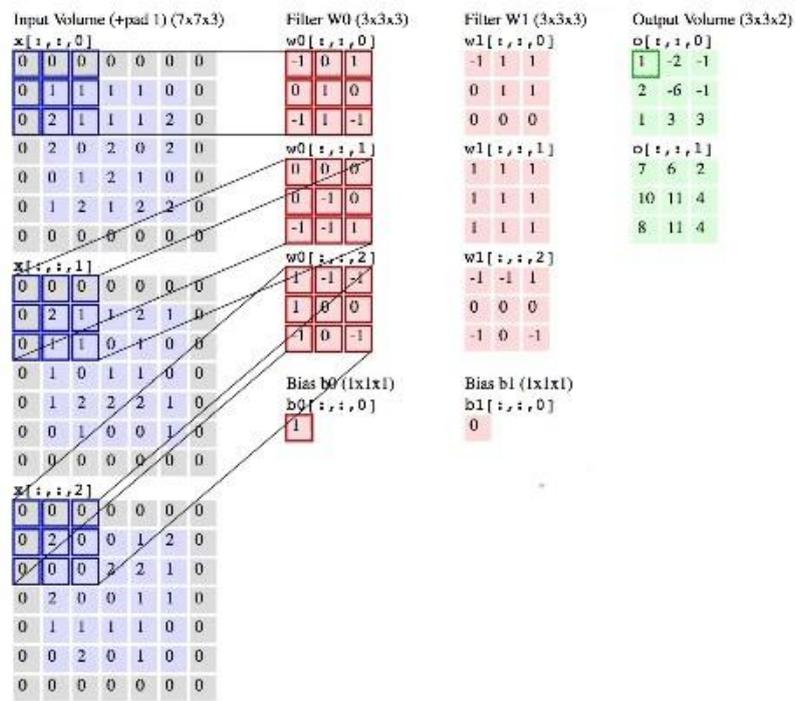
Gambar 2. 3 Citra RGB

Sumber : (<https://medium.com/@16611110/apa-itu-convolutional-neural-network-836f70b193a4>)

Gambar diatas menunjukkan gambar *RGB* berukuran 32x32 Pixel dengan jumlah *channel* nya adalah sebanyak 3 buah yang sebenarnya adalah *multidimensional array* Nantinya *Input Layer* akan menampung nilai pixel dari gambar yang akan menjadi input.

B. *Convolution Layer*

Convolutional layer merupakan layer yang paling penting pada CNN. Lapisan terdiri dari jaringan *neuron* yang tersusun sedemikian rupa hingga membentuk sebuah *filter*, lapisan ini juga menghasilkan gambar yang baru yang berisi *features* dari gambar yang di input pada *Input Layer* sebelum nya. *Filter* yang di implementasi berupa matriks berukuran 1x1 atau 5x5. Setelah di *filter*, hasil dari filter tersebut adalah *feature map* yang akan di gunakan pada *Activation Layer*.



Gambar 2. 4 Convolution Layer

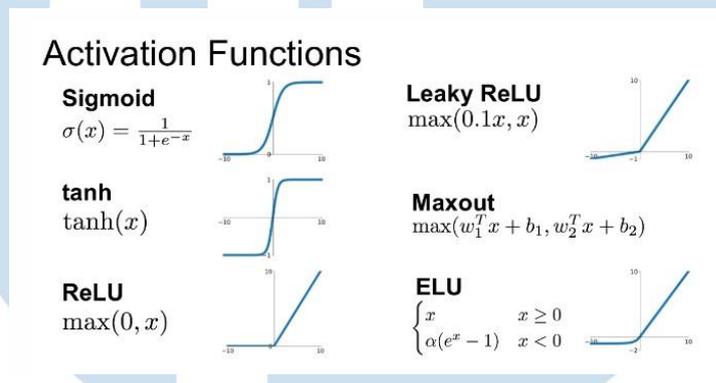
Sumber : (<https://medium.com/@16611110/apa-itu-convolutional-neural-network-836f70b193a4>)

Gambar di atas menunjukkan proses pemfilteran dengan matriks 5x5 dan memiliki *channel* 3. Ketiga filter tersebut akan bergeser pada semua bagian gambar. Pada pergeseran tersebut akan dilakukan operasi perkalian “dot” antara input dan nilai filter sehingga menghasilkan Feature Map.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

C. Activation Layer

Sebelum masuk kedalam *pooling layer*, gambar akan masuk ke dalam *activation* terlebih dahulu. Pada *layer* ini nilai *feature map* yang sudah dihasilkan dari *convolution layer* akan di pakai dan nilai tersebut akan di ubah tergantung *activation layer* yang akan di pakai. Beberapa *activation* yang biasa di pakai untuk CNN adalah sebagai berikut :

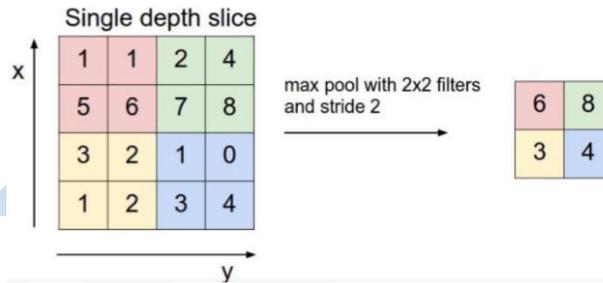


Gambar 2. 5 Jenis Activation Layer

Sumber : (<https://medium.com/@16611110/apa-itu-convolutional-neural-network-836f70b193a4>)

D. Pooling Layer

Pada *layer* ini terjadi proses mengurangi resolusi pada suatu gambar, namun informasi penting tetap di pertahankan (*downsampling*). Di lakukan *downsampling* agar proses komputasi berjalan lebih cepat. Hal terpenting dalam pembuatan model CNN adalah dengan memilih banyak jenis lapisan pooling. Hal ini dapat menguntungkan kinerja model (Lee, Gallagher, & Tu, 2015). Lapisan *pooling* berukuran filter 2x2 biasanya paling umum di gunakan. Berikut contoh *pooling layer*.

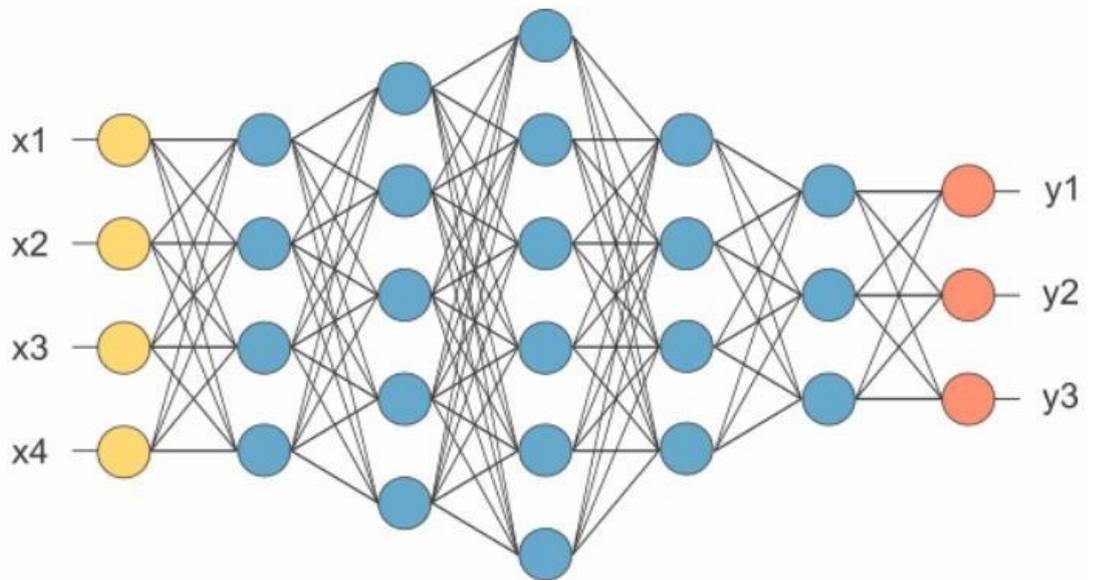


Gambar 2. 6 Pooling Layer

Sumber : (<https://medium.com/@16611110/apa-itu-convolutional-neural-network-836f70b193a4>)

E. Fully Connected Layer

Fully Connected Layer merupakan lapisan dimana semua *neuron* dari proses sebelum nya terhubung dengan lapisan *neuron* selanjutnya. Setiap hasil proses dari layer sebelum nya diubah menjadi data 1 dimensi sebelum dapat di hubungkan ke semua neuron yang ada.



Gambar 2. 7 Fully Connected Layer

Sumber : (<https://medium.com/@16611110/apa-itu-convolutional-neural-network-836f70b193a4>)

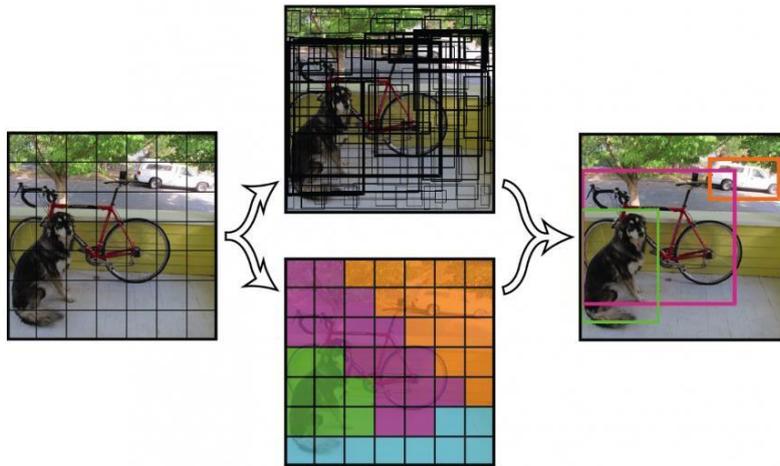
2.2.2 OpenCV

OpenCV merupakan library yang bersifat open source yang di kembangkan oleh intel. Intel pertama kali meluncurkan versi OpenCV pada tahun 1999. Awalnya untuk menggunakan OpenCV di perlukan library dari *Intel ImageProcessing Library*. Kemudian akhirnya di hilangkan dan OpenCV menjadi library standalone seperti sekarang ini. OpenCV memudahkan programmer untuk mengolah suatu citra digital. Fitur yang ada di dalam OpenCV meliputi pengenalan wajah, pelacakan wajah, deteksi wajah, kalman filtering dan AI. Dengan OpenCV programmer memungkinkan untuk memproses image dan video sesuai dengan kebutuhan dan juga dapat melakukan eksperimen uji dengan berbagai algoritma *Computer Vision*.

2.2.3 YOLOv3 (You Only Look Once)

YOLO merupakan algoritma *computer vision* yang cukup populer. Yolo menerapkan jaringan syaraf tunggal pada keseluruhan bagian gambar. Lalu jaringan syaraf tunggal tadi akan di bagi – bagi menjadi wilayah dan kemudian memprediksi kotak pembatas serta probabilitasnya untuk setiap wilayah pembatas agar dapat mengklasifikasi suatu objek.





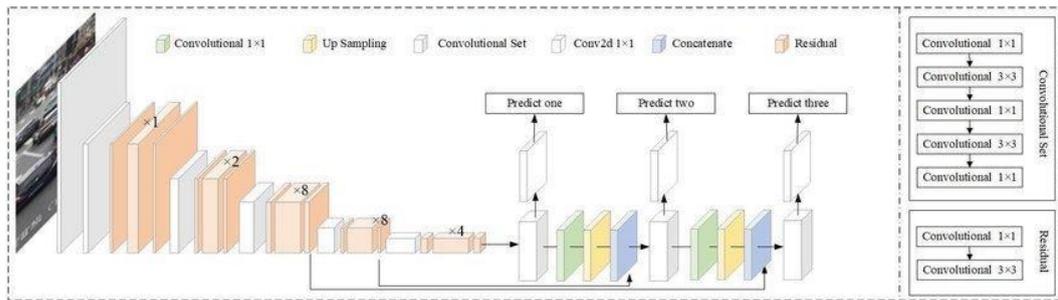
Gambar 2. 8 YOLO v3

Sumber: (Redmon et al., 2016)

Dapat kita lihat pada gambar terakhir pada gambar 4.5 kotak pembatas dipilih dengan nilai yang paling tinggi sebagai pemisah antara objek satu dengan objek lain nya. YOLO mempunyai beberapa versi, yaitu YOLOv1, YOLOv2 dan YOLOv3.

Dahulu YOLOv2 merupakan algoritma yang cepat dan akurat, namun sayangnya YOLOv2 masih menggunakan 19 *layer network* dan 11 *network* lagi untuk mendeteksi objek. Dengan arsitektur 30 *layer* tersebut YOLOv2 mengalami kesulitan untuk mendeteksi objek –objek kecil. Dan YOLO v2 sayangnya tidak memiliki elemen yang penting seperti *residual block*, tidak ada *skip connection* dan tidak ada *upsampling*. Maka dari permasalahan YOLOv2 tersebut, YOLOv3 di kembangkan.

YOLOv3 menggabungkan semua yang tidak ada di YOLOv2 sebelumnya. YOLOv2 memiliki 53 *layer network* yang dilatih menggunakan *Imagenet*. 53 *layer* tersebut di tumpuk lagi dengan *layer* yang berjumlah 53, sehingga total dari *layer* nya adalah 106 *layer*. Jadi pada YOLOv3 terdapat 106 *Layer Fully Convolutional*. Dikarenakan *layer* yang banyak tersebut YOLOv3 lebih lambat di banding dengan YOLOv2.



Gambar 2. 9 Arsitektur YOLO v3
 Sumber: (Redmon et al., 2016)

Algoritma YOLOv3 pada proses pertama akan dilakukan memisahkan gambar menjadi kisi-kisi. Setiap sel *grid* memprediksi sejumlah kotak batas di sekitar objek yang mendapat skor tinggi dengan kelas yang telah ditentukan sebelumnya. Setiap kotak batas memiliki skor confidence masing-masing tentang seberapa akurat asumsi bahwa prediksi seharusnya dan hanya mendeteksi satu objek per kotak pembatas. Kotak batas dihasilkan dengan mengelompokkan dimensi kotak berdasarkan parameter dari dataset asli untuk menemukan bentuk dan ukuran yang paling umum. Terdapat algoritma lain yang dapat melakukan hal tersebut, yaitu RCNN, Fast RCNN. Namun YOLO dapat mengklasifikasi dan regresi kotak pembatas secara bersamaan.

2.2.4 Recall

Recall merupakan jumlah keseluruhan sampel positif yang dapat diklasifikasikan dengan benar kemudian di bagi dengan total jumlah positif. Jika hasil menunjukkan high recall berarti algoritma mendeteksi FN dengan jumlah sedikit. Rumus untuk menghitung recall adalah sebagai berikut :

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

2.2.5 Precision

Nilai *precision* di hitung dengan cara nilai true positif di bagi dengan *True positif* dan *False positif*. Jika presisi tinggi berarti jumlah false positif nya sedikit. Berikut rumus untuk menghitung *precision* :

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

2.2.6 F Score

F Score merupakan metrik yang digunakan untuk menghitung gabungan dari *precision* dan *recall*. Persamaan F score ditunjukkan pada di bawah ini :

$$F1\ Score = \frac{2 * Recall * Precision}{Recall + Precision}$$

2.2.7 Average Precision

Mean average precision merupakan nilai rata-rata dari average precision (AP) yang membentuk metrik evaluasi untuk mengukur kinerja dari sebuah deteksi objek. Nilai AP didapatkan dari perhitungan *precision*.

$$AP = \sum (Recall_{n+1} - Recall_n) \cdot Precision_{interp.}(Recall_{n+1})$$

U M M N
UNIVERSITAS
MULTIMEDIA
NUSANTARA