BAB II

LANDASAN TEORI

2.1 You Only Look Once (YOLO)

Algoritma *You Only Look Once* (YOLO) adalah algoritma pendeteksian objek *real-time*, yang tidak menghabiskan waktu begitu banyak untuk menghasilkan kawasan pendeteksian objek (Nishad, 2019). YOLO menggunakan pendekatan jaringan syaraf tiruan (JST) untuk mendeteksi objek pada sebuah citra. Jaringan ini akan membagi gambar menjadi wilayah-wilayah yang memungkinkan pengklasifikasian sebagai objek yang dimaksud atau bukan (Yanuar, 2018). Algoritma ini sangat cepat yaitu 45 *frame* per-detik, bahkan algoritma ini diperbarui sehingga kecepatannya 155 *frame* per-detik (Redmon dkk, 2015). Algoritma *You Only Look Once* (YOLO) membagi gambar input menjadi kotak S x S. Jika suatu objek terdapat di dalam kotak prediksi maka, kotak prediksi tersebut yang akan bertanggung jawab untuk mendeteksi objek tersebut (Chablani, 2017).

Manfaat menggunakan algoritma *You Only Look Once* (YOLO) menurut (Redmon dkk, 2015).

1. Algoritma YOLO sangat cepat. Karena algoritma ini membagi wilyah-wilayah menjadi kotak-kotak untuk memisahkan objek dan wilayah lainnya.

2. Algoritma YOLO melihat seluruh gambar pada proses *training* dan pengujian tidak seperti algoritma pada umumnya yang melakukan beberapa iterasi untuk memproses suatu gambar.
3. Algoritma YOLO mempelajari representasi objek yang dapat digeneralisasikan.

Jaringan YOLO menggunakan fitur dari seluruh gambar untuk memprediksi setiap kotak pembatas. Desain YOLO memungkinkan pelatihan dari ujung ke ujung dengan waktu yang cepat dengan tetap mempertahankan presisi rata-rata yang tinggi. Berikut ini adalah langkah-langkah cara kerja algoritma YOLO menurut (Rahyagara, 2018).

1. Dataset

   Digunakan untuk meyimpan informasi-informasi terkait objek yang akan dideteksi.

2. Anotasi citra

   Proses pembuatan label dengan memberikan (*bounding box*) berserta memberikan nama kelas pada objek setiap citra.

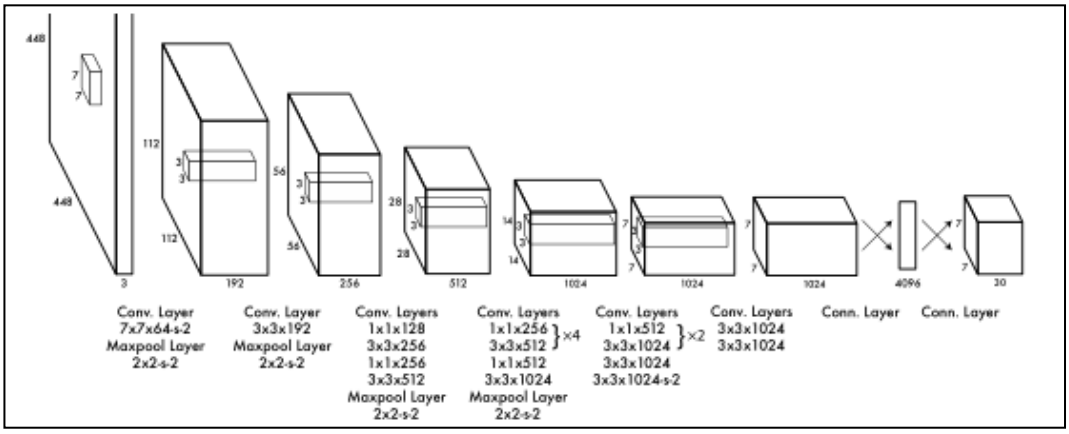
3. Training

   Pada proses training yang harus penting dipahami

   - *Batch size* adalah banyaknya data yang digunakan untuk setiap pelatihan, semakin banyak daya semakin baik hasilnya.
   - *Epoch* adalah banyaknya proses iterasi.
   - *Learning rate* adalah seberapa cepat proses pelatihan, semakin cepat *learning rate* maka semakin mudah untuk mendeteksi objek.

Algoritma *You Only Look Once* (YOLO) untuk saat ini mempunyai 3 versi yaitu YOLOv1, YOLOv2/YOLO9000 dan YOLOv3 (Jonnalagadda ,2019).

a. YOLOv1



Gambar 2.1 Arsitektur YOLOv1 (Redmon dkk, 2015)

YOLOv1 menggunakan kerangka kerja Darknet yang dilatih pada ImageNet-1000. YOLOv1 tidak dapat berfungsi secara maksimal untuk pendeteksian objek yang kecil. Arsitektur ini kesulitan dalam generalisasi objek jika gambar dari dimensi berbeda dari gambar yang dilatih (Jonnalagadda, 2019).

b. YOLOv2/YOLO9000

Type	Filters	Size/Stride	Output
Convolutional	32	3 × 3	224 × 224
Maxpool		2 × 2/2	112 × 112
Convolutional	64	3 × 3	112 × 112
Maxpool		2 × 2/2	56 × 56
Convolutional	128	3 × 3	56 × 56
Convolutional	64	1 × 1	56 × 56
Convolutional	128	3 × 3	56 × 56
Maxpool		2 × 2/2	28 × 28
Convolutional	256	3 × 3	28 × 28
Convolutional	128	1 × 1	28 × 28
Convolutional	256	3 × 3	28 × 28
Maxpool		2 × 2/2	14 × 14
Convolutional	512	3 × 3	14 × 14
Convolutional	256	1 × 1	14 × 14
Convolutional	512	3 × 3	14 × 14
Convolutional	256	1 × 1	14 × 14
Convolutional	512	3 × 3	14 × 14
Maxpool		2 × 2/2	7 × 7
Convolutional	1024	3 × 3	7 × 7
Convolutional	512	1 × 1	7 × 7
Convolutional	1024	3 × 3	7 × 7
Convolutional	512	1 × 1	7 × 7
Convolutional	1024	3 × 3	7 × 7
Convolutional	1000	1 × 1	7 × 7
Avgpool		Global	1000
Softmax			

Gambar 2.2 Arsitektur YOLOv2/YOLO9000 (Jonnalagadda, 2019)

YOLOv2 menggunakan arsitektur Darknet-19 dengan 19 *convolutional layers*, 5 *max pooling layers* dan *softmax layer* untuk klasifikasi objek. Darknet adalah kerangka kerja jaringan saraf yang ditulis dalam Clanguage dan CUDA. Arsitektur ini sangat cepat dalam pendeteksian objek secara *real-time* (Jonnalagadda, 2019).

c. YOLOv3

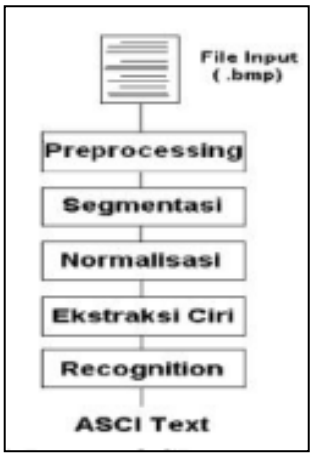
	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Gambar 2.3 Arsitektur YOLOv3 (Jonnalagadda, 2019)

YOLOv3 menggunakan arsitektur Darknet-53 yang memilki 53 *convolutional layers*. Arsitektur ini jauh lebih dalam dari pada YOLOv2 dan juga memiliki *shotcut connections*. Darknet 53 terdiri dari 3x3 dan 1x1 *filters* dengan *shortcut connections* (Jonnalagadda, 2019).

2.2 Optical Character Recognition (OCR)

Optical Character Recognition adalah proses konversi gambar (huruf) menjadi karakter ASCII yang dapat dikenali oleh komputer. Optical Character Recognition (OCR) adalah sistem yang dikembangkan pada tahun 1914 oleh Emanuel Goldberg yang awalnya berfungsi untuk telegram dan alat baca untuk orang tunanetra. Saat ini, kemajuan OCR terus dikembangkan sehingga dapat menghasilkan akurasi yang lebih baik untuk karakter yang sulit untuk dikenali (Phangtriastu, 2017). Gambar huruf berupa hasil scan dokumen, hasil print-screen halaman *web*, hasil foto, dan lain-lain (Mohammad dkk, 2014). Sistem OCR terdiri dari banyak fase. Berikut adalah fase dari pemrosesan OCR (Hartanto dkk, 2014).



Gambar 2.4 *Stage of* OCR (Hartanto, dkk, 2014).

1. ***Preprocessing***

Pada proses ini gambar akan dikonversi ke skala abu-abu. Gambar skala abu-abu dikonversi ke dalam gambar biner.

2. ***Segmentation***

Segmentasi citra biner bertujuan untuk mengelompokkan pixel-pixel objek mendjadi wilayah yang merepresentasi suatu objek. Batas antara objek dengan latar belakang terlihat jelas pada citra biner. Pixel objek berwarna hitam sedang pixel latar belakang berwarna putih.

3. **Normalisasi**

Normalisasi merupakan salah satu tahap dari *preprocessing* citra yang dilakukan sebelum masuk ke proses pengenalan. Proses ini bertujuan untuk menyesuaikan data citra masukan dengan data citra pada basis data.

4. **Ekstraksi Ciri**

Tujuan dari proses ini adalah untuk mendeteksi karakteristik yang paling penting dari karakter, dan secara umum bahwaini adalah masalah yang paling sulit dalam pengenalan pola. Cara terbaik untuk menggambarkan karakter adalah dengan gambar aktual.

5. ***Recognition***

Pengelompokkan karakter untuk membuat karakter dikenali, pendeteksian kesalahan dan akan dilakukannya pengoreksian. Berikut ini adalah dokumen gambar yang telah dideteksi teks dan dipisahkan dari *background*.

2.3 User Time

Pada penelitian ini, dibutuhkan menghitung waktu eksekusi untuk mengevaluasi performa dari suatu algoritma. Dalam penelitian menggunakan metode *user time* dalam pendeteksian logo menggunakan algoritma *You Only*

Look Once (YOLO). Waktu ekseskusi dibagi menjadi 4 bagian: *simulated time*, *user time*, *system time* dan *elapsed time* (Wicaksana dan Tang, 2017). *Simulated time* adalah durasi waktu simulasi secara berjalan bersamaan (Wicaksana dan Tang, 2017). *User time* adalah *wall-clock time* yang diterapkan pada simulator *System time* adalah waktu yang digunakan untuk melakukan tugas sistem atas nama simulator (Wicaksana dan Tang, 2017). *Elapsed time* adalah waktu yang dicocokan dengan jam dinding dari proses simulasi sampai waktu statistik simulasi dicetak (Wicaksana dan Tang, 2017).

2.4 Format Sampul Proposal

Pada penelitian ini, pengecekan format sampul proposal Skripsi antara lain adalah pengecekan kesesuai kode penelitian dengan nama bidang penelitian dan minimal 1 kode penelitian pada sampul proposal Skripsi, pengecekan judul proposal Skripsi dengan format maksimal 15 kata (tidak termasuk penggunaan tempat implementasi atau studi kasus) dan yang terakhir halaman sampul depan harus dicantumkan data lembaga meliputi nama Program Studi, Fakultas, Perguruan Tinggi, tempat/kota Perguruan Tinggi dan tahun pembuatan proposal Skripsi (Meidia dan dkk, 2019). Contoh format sampul dapat dilihat pada Gambar 2.2.

LAMPIRAN 2: Contoh Halaman Sampul Depan dan Halaman Judul

JUDUL SKRIPSI
(ukuran: 14 pt Times New Roman)

SKRIPSI
(ukuran: 14 pt Times New Roman)

Diajukan sebagai salah satu syarat untuk memperoleh gelar
Sarjana Komputer (S.Kom.)
(ukuran: 12 pt Times New Roman)

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

<Nama Mahasiswa>
<NIM>
(ukuran: 12 pt Times New Roman)

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK DAN INFORMATIKA
UNIVERSITAS MULTIMEDIA NUSANTARA
TANGERANG
<Tahun>
(ukuran: 12 pt Times New Roman)

Gambar 2.5 Gambar Format Sampul