

BAB II LANDASAN TEORI

2.1 Penelitian Terhadulu

2.1.1 *A Lightweight Model for Detecting Forest Fire Smoke in UAV Images Based on YOLOv7* [3]

Pada penelitian ini penulis membuat sebuah sistem untuk mendeteksi asap kebakaran hutan untuk mencegah kebakaran sejak awal. Pada penelitian ini model yang digunakan adalah YOLOv7 penulis menggunakan model tersebut karena YOLOv7 adalah versi terbaru dari YOLO yang memiliki kemampuan deteksi lebih baik dari pada versi YOLO sebelumnya. Selain itu mampu mendeteksi secara cepat dan akurasi lebih tinggi. Jumlah data yang digunakan pada penelitian ini sebanyak 5311 yang berisi gambar asap kebakaran hutan, asap kecil, asap dengan *distractor*, dan asap sintesis dengan sudut pandang yang berbeda. Data tersebut dibagi menjadi 4249 data *training*, 531 data *validation*, dan 531 data *testing*. Parameter yang dipakai dalam penelitian ini adalah *epoch* 300, *batch* 16, *image size* 640x640, dan *optimizer* SGD. Hasil yang didapatkan dalam penelitian ini adalah model YOLOv7 mampu mendeteksi asap kebakaran hutan dengan nilai akurasi sebesar 80.2%, *precision* 77.1%, dan *recall* 71.8%.

Terdapat beberapa poin penting yang dapat diambil antara lain :

- Model YOLO dapat digunakan untuk deteksi asap kebakaran hutan.
- YOLOv7 mampu mendeteksi secara cepat dan akurasi lebih tinggi.
- Metode YOLOv7 adalah versi terbaru dari YOLO.

2.1.2 Fire Detection and Notification Method in Ship Areas Using Deep Learning and Computer Vision Approaches [4]

Pada penelitian ini, penulis membuat sistem yang mampu mendeteksi kebakaran yang terjadi di atas kapal. Metode yang digunakan dalam penelitian ini adalah metode YOLOv7. Penulis menggunakan metode tersebut karena YOLOv7 mampu mendeteksi objek-objek kecil sehingga cocok digunakan untuk melakukan deteksi kebakaran di kapal yang sering kali terlokalisasi. Hasil yang didapatkan dalam penelitian ini adalah model YOLOv7 mampu mendeteksi objek kebakaran kapal dengan akurasi sebesar 93%. Terdapat beberapa poin penting yang dapat diambil antara lain :

- Metode YOLOv7 mampu mendeteksi objek kecil.

2.1.3 An Efficient Forest Fire Detection Based on Improved YOLOv5 [5]

Pada penelitian ini penulis membuat sistem untuk mendeteksi kebakaran hutan. Sistem ini dibuat untuk memperkirakan terjadinya kebakaran hutan dan mendeteksi kebakaran pada tahap awal. Pada penelitian ini metode yang dipakai adalah YOLOv5. Penulis menggunakan metode tersebut karena YOLOv5 dapat menangani data berskala besar dan mendukung deteksi objek secara real time. Selain itu metode YOLOv5 mampu mengenali objek pada ukuran yang berbeda-beda. Jumlah data yang digunakan sebanyak 2328 gambar dengan 716 gambar kebakaran hutan dan 1612 gambar hutan tanpa kebakaran. Parameter yang digunakan dalam penelitian ini yaitu dengan *image size* 640x640, *200 epoch*, *batch size* 8, *optimizer* SGD. Hasil yang didapatkan dalam penelitian ini adalah metode YOLOv5 mampu mendeteksi kebakaran hutan dengan akurasi sebesar 85.67%, *precision* 79.88%, dan *recall* 80.55%. Terdapat beberapa poin penting yang dapat diambil antara lain :

- Metode YOLOv5 dapat menangani data berskala besar dan mendukung deteksi objek secara *real time*.

- Metode YOLOv5 mampu mengenali objek pada ukuran yang berbeda beda.

2.1.4 *Early Wildfire Smoke Detection Using Different YOLO Models* [6]

Pada penelitian ini penulis membuat sistem pendeteksi asap kebakaran hutan. Tujuan dari penelitian ini adalah untuk mendeteksi kebakaran dan asap pada tingkat yang memuaskan. Data yang digunakan sebanyak 16.503 foto. Metode yang digunakan dalam penelitian ini adalah YOLOv3, YOLOv5, YOLOv7. Dataset yang digunakan berasal dari Kaggle dengan total 1723 titik data. Dataset dibagi menjadi 80 data untuk pelatihan dan 20 data untuk pengujian. Ukuran gambar yang digunakan adalah 640x640 dengan 100 *epoch* dan 16 *batch*. Model YOLO memiliki 29 parameter berbeda untuk penyetelan *hyperparameter*. Pencarian ini menggunakan dua belas parameter yaitu *lr0*, *lrf*, *momentum*, *box loss gain*, *classification loss gain*, *object loss gain*, *IoU training* ambang batas, *optimizer* (SGD), *output* jangkar pada tiap layer, ukuran gambar *input*, *batch*, *epoch*. Hasil yang didapatkan pada penelitian ini adalah metode YOLOv5 mendapatkan akurasi sebesar yaitu 96.8% kemudian YOLOv7 dengan akurasi sebesar 95.08% dan 94.8% dengan model YOLOv3 seluruh akurasi tersebut didapatkan pada IoU 50%.

Terdapat beberapa poin penting yang dapat diambil antara lain :

- YOLOv5 mendapatkan akurasi lebih tinggi daripada YOLOv7 dan YOLOv3.
- Setiap model mempunyai Tingkat kecepatan deteksi yang berbeda beda.
- YOLOv7 melakukan deteksi lebih lambat dari pada YOLOv5.

2.1.5 *A Wildfire Smoke Detection System Using Unmanned Aerial Vehicle Image Based on the Optimized YOLOv5* [7]

Pada penelitian ini penulis membuat sistem pendeteksi asap kebakaran hutan. Tujuan dari penelitian ini adalah untuk mendeteksi keberadaan api dari asap yang muncul. Metode yang digunakan dalam

penelitian ini adalah YOLOv5. Dataset yang digunakan memiliki total 6000 data yang terdiri dari 3285 data asap kebakaran dan 2715 tanpa asap kebakaran. Ukuran gambar yang digunakan adalah 640x640 dengan 300 *epoch*. Pada penelitian ini juga dilakukan proses augmentasi rotasi untuk menambah variasi data yang digunakan. Hasil yang didapatkan pada penelitian ini adalah metode YOLOv5 mampu mendeteksi gambar dengan baik dengan bantuan augmentasi rotasi yang mendapatkan nilai akurasi sebesar 87.2%.

Terdapat beberapa poin penting yang dapat diambil antara lain :

- Augmentasi mampu membantu meningkatkan kinerja model.

2.2 Tinjauan Teori

2.2.3 *Object Detection*

Object Detection adalah proses untuk menentukan keberadaan sebuah objek pada suatu citra digital. Tujuan deteksi objek untuk menemukan semua bentuk atau beberapa golongan objek tertentu.

Deteksi objek adalah proses identifikasi dan lokalisasi objek tertentu dalam gambar atau video. Dalam konteks kecerdasan buatan, deteksi objek melibatkan penggunaan algoritma dan teknik *computer vision* untuk mengenali dan memisahkan objek-objek yang berbeda dalam sebuah gambar atau video. Tujuan dari deteksi objek adalah untuk mengidentifikasi dan melacak objek-objek yang penting atau menarik perhatian dalam suatu konteks tertentu.

2.2.2 *Neural Network*

Neural network adalah jaringan saraf tiruan yang terinspirasi oleh struktur dan fungsi otak manusia. *Neural network* terdiri dari banyak unit pemrosesan kecil yang disebut neuron, yang saling terhubung. Neuron-neuron ini bekerja sama untuk memproses informasi dan membuat keputusan.

Neural network dapat digunakan untuk berbagai macam tugas, di antaranya untuk klasifikasi gambar, pengenalan objek, terjemahan bahasa, pemrosesan bahasa alami, pengendalian robot, dan melakukan prediksi.

Neural network dilatih dengan cara memberikan *input* data dan *output* yang diinginkan. *Neural network* kemudian akan menyesuaikan bobotnya

sehingga output yang dihasilkan sesuai dengan *output* yang diinginkan. Proses pelatihan ini dapat dilakukan berulang kali sampai *neural network* dapat memproses data dengan akurat.

Neural network telah menjadi alat yang sangat penting dalam banyak bidang, termasuk pembelajaran mesin, *deep learning*, dan kecerdasan buatan. *Neural network* dapat digunakan untuk berbagai macam tugas dan telah terbukti dapat mencapai hasil yang sangat baik.

2.2.3 Augmentasi Data

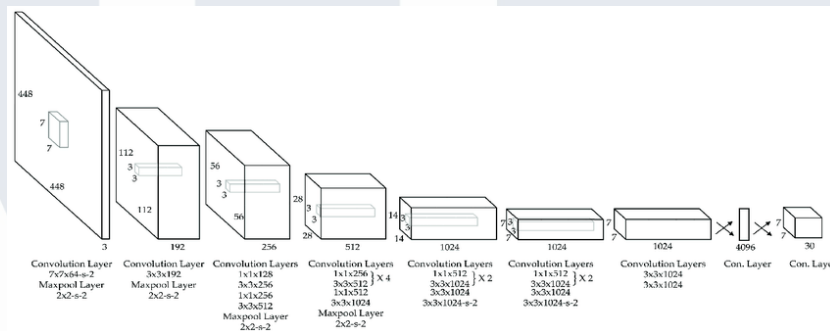
Augmentasi data adalah proses memodifikasi atau memperluas kumpulan data yang ada dengan melakukan perubahan pada data yang sudah ada. Dalam konteks pemrosesan gambar atau pengenalan pola, augmentasi sering digunakan untuk menambah jumlah sampel data pelatihan dengan membuat variasi pada gambar yang ada. Tujuannya adalah untuk meningkatkan keragaman data dan menghindari *overfitting* pada model pembelajaran mesin. Contoh penyempurnaan gambar dapat mencakup memutar, memindahkan, memotong, menskalakan, mengubah kecerahan, dan menerapkan efek lainnya. Dengan melakukan penskalaan, model pembelajaran mesin dapat mempelajari variasi yang lebih besar dan menjadi lebih baik dalam mengenali objek atau pola dalam data baru.

2.2.4 YOLO

YOLO (*You Only Look Once*) adalah metode deteksi objek *real-time* yang sangat populer dalam bidang pengolahan citra dan pengenalan objek. Metode ini dikembangkan dengan tujuan untuk menghasilkan deteksi objek yang cepat dan akurat.

You Only Look Once (YOLO) adalah model yang menggunakan *Region based Convolutional Neural Networks* (R-CNN) untuk mendeteksi objek. Model ini dapat melakukan deteksi beberapa objek, memprediksi kelas yang dihasilkan, dan menentukan lokasi objek. YOLO dianggap sebagai arsitektur yang sangat cepat dan akurat. Namun keakuratan arsitektur ini dipengaruhi oleh beberapa variabel [4].

YOLO menggabungkan komponen terpisah dari deteksi objek ke dalam jaringan saraf tunggal dan menggunakan fitur dari seluruh gambar untuk memprediksi setiap kotak pembatas, memungkinkan pelatihan kehadiran sementara. Analisis *real-time* cepat dan komprehensif dengan tetap mempertahankan tingkat akurasi rata-rata yang tinggi. YOLO versi terbaru dirilis, tepatnya YOLO versi tujuh pada Juli 2022, yang diklaim mengungguli semua pendeteksi objek dalam hal kecepatan dan akurasi antara 5 FPS hingga 160 FPS. [7].



Gambar 2.1 Arsitektur YOLO

Dilihat dari arsitekturnya, model ini memiliki total 24 *convolutional layer*, 4 *max-pooling layer*, dan 2 *fully connected*. Jika dijelaskan, setiap lapisan akan beroperasi sebagai berikut:

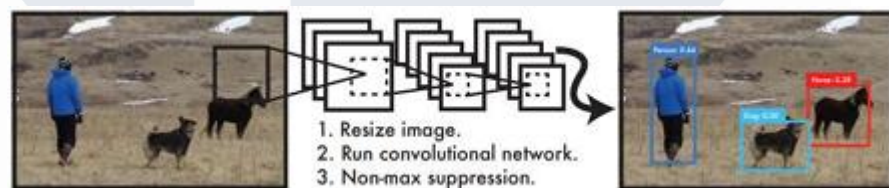
- Pada lapisan pertama, gambar dimasukkan dan akan diubah ukurannya menjadi resolusi 448x448 sebelum masuk ke *convolutional network*.
- Selanjutnya, konvolusi 1x1 diterapkan untuk mengurangi jumlah saluran, diikuti konvolusi 3x3 untuk menghasilkan keluaran kubus.
- ReLU (*Rectified Linear Unit*) digunakan untuk fungsi aktivasi, kecuali lapisan terakhir yang menggunakan fungsi aktivasi linier.
- Dan beberapa teknik tambahan juga digunakan seperti normalisasi batch dan imputasi, yang masing-masing memiliki efek menyetel model dan mencegah *overfitting*.

Gambar di atas adalah arsitektur dari metode YOLO. YOLO telah mengalami beberapa iterasi pengembangan, seperti YOLOv1, YOLOv2, YOLOv3, dan YOLOv4, YOLOv5, YOLOv6 dan YOLOv7, yang terus meningkatkan kecepatan dan akurasi deteksi objek.

2.2.5 YOLOv5

YOLOv5 menggunakan metode jaringan saraf tiruan untuk mendeteksi objek dalam gambar. Membagi gambar menjadi beberapa wilayah dan prediksi setiap kotak pembatas beserta probabilitas untuk setiap wilayah. Kotak pembatas ini kemudian dibandingkan dengan setiap probabilitas yang diprediksi [10].

Konsep algoritma YOLOv5 sangat sederhana. Gambar di bawah menjelaskan konsep sederhana YOLO. Pertama, sistem mengubah ukuran gambar masukan menjadi 448 x 448 piksel (seperti yang diharapkan dalam implementasi saat ini). Sistem kemudian menjalankan jaringan konvolusional tunggal pada gambar. Pada akhirnya, sistem akan memungkinkan deteksi objek dengan tingkat kepercayaan bergantung pada model yang dilatih.[11].



Gambar 2.2 Sistem Pendeteksi YOLOv5

Model YOLOv5 memiliki beberapa kelebihan dari model objek deteksi lainnya. Berikut adalah kelebihan model algoritma YOLOv5[11] :

1. Proses deteksi di YOLOv5 sangat cepat. YOLOv5 menggunakan model pelatihan sebagai acuan deteksi gambar uji baru sehingga tidak perlu dilatih berulang kali untuk mendeteksi gambar baru. Jaringan YOLOv5 dapat mencapai akurasi deteksi dua kali lipat dibandingkan sistem *real-time* lainnya.
2. YOLOv5 memeriksa keseluruhan gambar dan fitur-fiturnya selama pelatihan dan pengujian sehingga YOLOv5 secara implisit menyajikan informasi yang kontekstual tentang kelas dan gambar. Dengan beberapa keunggulan tersebut, YOLOv5 tetap memiliki kelemahan di mana YOLOv5 masih tertinggal di sisi akurasi. Namun itu yang menjadi perkembangan untuk setiap versi YOLOv5 ke depannya.

2.2.6 YOLOv7

YOLOv7 adalah versi terbaru dari YOLO. YOLOv7 melampaui semua detektor objek yang dikenal baik dalam kecepatan maupun akurasi dalam kisaran 5 FPS hingga 160 FPS, dan memiliki akurasi tertinggi pada 56,8% AP di antara semua detektor objek *real-time* yang dikenal dengan 30 FPS atau lebih tinggi pada *Graphics Processing Unit* (GPU) V100. YOLOv7 meningkatkan akurasi deteksi objek waktu nyata tanpa meningkatkan biaya inferensi; mengurangi sekitar 40% parameter dan 50% komputasi detektor objek *real-time* yang canggih, serta memiliki kecepatan inferensi yang lebih cepat dan akurasi deteksi yang lebih tinggi [3].

YOLOv6 juga menawarkan peningkatan besar dalam hal mendeteksi tetapi kurang skalabilitas jika dibandingkan dengan YOLOv5 dan YOLOv7. Selain itu, kinerja YOLOv6 lebih akurat saat digunakan untuk gambar Tunggal inferensi dibandingkan dengan akurasi inferensi beberapa gambar yang ditawarkan oleh YOLOv5 dan YOLOv7 [3]. Sebagai hasilnya, percobaan dilakukan dengan YOLOv5 dan YOLOv7 karena keduanya cocok untuk beberapa objek deteksi, memberikan kemudahan dalam menyesuaikan proses *training* dan menjalankan inferensi.

2.2.7 Confusion Matrix

Confusion matrix juga sering disebut *error matrix*. Pada dasarnya *confusion matrix* memberikan informasi perbandingan hasil klasifikasi yang dilakukan oleh sistem (model) dengan hasil klasifikasi sebenarnya. Dari tabel yang didapatkan, untuk model klasifikasi yang dimiliki dapat dihitung akurasi, presisi, F-Score dan masih banyak variabel yang bisa dihitung berdasarkan kondisi data yang diprediksi atau diklasifikasikan. Isi dari tabel *confusion matrix* ada empat yang biasanya direpresentasikan dengan tabel seperti pada gambar 2.3 [5].

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Gambar 2.3 Confusion Matrix

- *True Positive (TP)*, Merupakan data positif yang diprediksi benar (TRUE) dan jawaban aktualnya adalah ya (TRUE).
- *True Negative (TN)*, Merupakan data negatif yang diprediksi benar (FALSE) dan jawaban aktualnya adalah tidak (FALSE).
- *False Positive (FP)*, Merupakan data negatif namun diprediksi sebagai data positif (TRUE) dan jawaban aktualnya adalah tidak (FALSE).
- *False Negative (FN)*, Merupakan data positif namun diprediksi sebagai data negatif (FALSE) dan jawaban aktualnya adalah ya (TRUE).

2.2.8 Accuracy

Accuracy menggambarkan seberapa akurat model dapat mengklasifikasikan dengan benar. Akurasi dapat dihitung menggunakan persamaan berikut.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

2.2.9 mAP

Mean Average Precision (mAP) adalah metrik yang digunakan untuk mengukur akurasi detektor objek di semua kelas dalam basis data tertentu. mAP hanyalah AP rata-rata di semua kelas yang dapat di formulakan seperti pada persamaan (6) berikut.

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i$$

Di mana AP_i adalah AP untuk kelas ke- i dan N adalah jumlah total kelas yang dievaluasi.

2.2.10 Precision

Presisi merupakan perbandingan jumlah data yang kategori positif yang diklasifikasikan secara benar oleh sistem dan keseluruhan data yang terklasifikasi positif. Presisi dapat dihitung menggunakan persamaan (4) berikut.

$$Precision = \frac{TP}{TP + FP}$$

2.2.11 Recall

Recall merupakan penggambaran keberhasilan model dalam menemukan kembali sebuah informasi. *Recall* dapat dihitung menggunakan persamaan (5) berikut.

$$Recall = \frac{TP}{TP + FN}$$

2.2.12 F1-Score

F1-Score menggambarkan perbandingan rata-rata *precision* dan *recall* yang dibobotkan. *Accuracy* tepat kita gunakan sebagai acuan performansi algoritma jika dataset kita memiliki jumlah data *False Negatif* dan *False Positif* yang sangat mendekati (*symmetric*). Namun jika jumlahnya tidak mendekati, maka sebaiknya kita menggunakan *F1-Score* sebagai acuan. *F1-score* menggunakan persamaan (6) berikut.

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

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