Perancangan Sistem Deteksi Api Menggunakan Framework YOLOv4

Made Radikia Prasanta[1], Muhammad Yoga Pranata[2]
[1,2]Program Studi Teknik Elektro, Fakultas Teknik, Universitas Negeri Malang
Jl. Semarang No. 5, Kec. Lowokwaru, 65145
made.radikia.1805366@students.um.ac.id

Abstrak

Kebakaran hutan dan lahan berdampak pada rusaknya ekosistem. Menurut data dari Kementerian Lingkungan Hidup dan Kehutanan (KLHK), luas hutan Indonesia per tahun 2013 mencapai 128,5 juta hektar. Ironisnya, hutan di Indonesia setiap tahunnya terus menyusut diakibatkan oleh kebakaran hutan. Merujuk pada perhitungan Ditjen Planologi KLHK, angka deforestasi Indonesia periode 2014-2015 mencapai 1,09 juta hektar dan 2015-2016 menjadi 0,63 juta hektar. Hal ini dapat dicegah dengan memanfaatkan UAV (*Unmanned Aerial Vehicle*) untuk melakukan pengamatan secara langsung melalui kamera. Penelitian ini dilakukan untuk mengembangkan sistem deteksi api menggunakan YOLOv4 dengan pengukuran kinerja sistem menggunakan *confusion matrix*. Algoritma YOLOv4 adalah algoritma *deep learning* yang menerapkan jaringan syaraf konvolusional (CNN). YOLOv4 yang diskalakan adalah jaringan neural paling akurat dan memelikiki kecepatan terhadap akurasi di seluruh rentang akurasi dan kecepatan dari 15 FPS hingga 1774 FPS. Hasil penelitian menunjukan framework YOLOv4 berhasil mendeteksi api dengan menghasilkan rata-rata akurasi sebesar 0,92. Akurasi terbaik didapat pada percobaan video 2 dengan skor *precision* sebesar 0,94, skor *recall* sebesar 0,99 dan skor *accuracy* sebesar 0,95.
Kata kunci: Deteksi Api, *Confusion Matrix*, YOLOv4.

I. PENDAHULUAN

Kebakaran hutan dan lahan berdampak pada rusaknya ekosistem dan menyebabkan musnahnya flora dan fauna yang tumbuh dan hidup di hutan. Dampak lainnya dari asap yang ditimbulkan dapat menyebabkan berbagai macam penyakit. Menurut data dari Kementerian Lingkungan Hidup dan Kehutanan, luas hutan Indonesia per tahun 2013 mencapai 128,5 juta hektar [1]. Ironisnya, hutan di Indonesia setiap tahunnya terus menyusut diakibatkan oleg kebakaran hutan. Merujuk pada perhitungan Ditjen Planologi KLHK, angka deforestasi Indonesia periode 2014-2015 mencapai 1,09 juta hektar dan 2015-2016 menjadi 0,63 juta hektar [2].

Salah satu solusi yang dapat dilakukan untuk mencegah terjadinya kebakaran hutan yaitu pemantauan kawasan hutan yang harus dilakukan secara berkala. Seperti penggunaan kamera yang dipasang pada batang pohon untuk mencakup luasan tertentu di hutan dalam mendeteksi dini kebakaran hutan, pendeteksian dilakukan secara realtime berbasis notifikasi menggunakan *raspberry* [3]. Selain itu pemantauan dapat dilakukan dengan memanfaatkan teknologi UAV (*Unmanned Aerial Vehicle*) atau pesawat tanpa awak yaitu pengamatan secara langsung melalui kamera yang ada di pesawat [4]. Dalam hal ini kamera pada pesawat bertujuan untuk mengambil foto atau video kondisi lingkungan atau hutan sekitar melalui udara [5], kamera terintegrasi dengan program komputer yang selanjutnya akan diproses untuk menentukan apakah terdapat api atau tidak.

Metode yang dapat dilakukan dalam mendeteksi api secara otomatis adalah dengan menggunakan pengolahan citra [6]. Dalam penggunaan teknik ini objek api dengan berbagai macam kondisi dapat secara mudah untuk dideteksi oleh sistem yaitu dengan menggunakan framework *You Only Look Once* (YOLO). YOLO adalah sebuah algoritma yang dikembangkan untuk mendeteksi sebuah objek secara realtime. Berbagai macam versi YOLO salah satunya yaitu YOLOv4. Algoritma YOLOv4 adalah algoritma *deep learning* yang menerapkan jaringan syaraf konvolusional (CNN). YOLOv4 yang diskalakan adalah jaringan neural paling akurat (55,8% AP) di kumpulan data Microsoft COCO dari semua jaringan neural [7]. YOLOv4 juga merupakan yang terbaik dalam hal rasio kecepatan terhadap akurasi di seluruh rentang akurasi kecepatan dari 15 FPS hingga 1774 FPS.

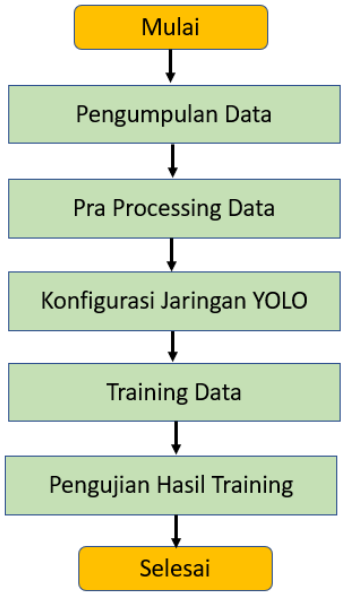
II. METODE PENELITIAN

A. Pengumpulan Data

Citra api yang digunakan dalam penelitian dikumpulkan, dijadikan sebagai dataset. Dataset api atau berupa berbagai macam gambar api seperti kebakaran hutan, lilin, kebakaran rumah dikumpulkan hingga berjumlah 2000 gambar. Gambar dikumpulkan dengan format jpg.

Gambar 1. Dataset Api

B. Rancangan Alur Kerja Sistem



Gambar 2. Skema Sistem

Alur kerja sistem diawali dengan pengumpulan data (akuisisi data) data dikumpulkan baik dari internet maupun hasil ekstraksi frame video perekaman api. Selanjutnya setiap gambar atau hasil ekstrasi akan mengalami perbaikan dengan mengubah *contrast stretching* pada tahapan pra processing data dengan melakukan pelabelan. Konfigurasi jaringan YOLO dilakukan untuk persiapan melakukan pelatihan data, kemudian citra yang sudah melalui pelabelan akan ditraining menggunakan model jaringan YOLOv4. Hasil training yang telah dilakukan yang dapat digunakan untuk mendeteksi api.

C. Pra Processing Data

Pada tahap ini data citra akan mengalami perbaikan kualitas citra dengan menggunakan metode *contrast stretching* atau peregangan kontras. *Contrast stretching* merupakan teknik untuk mendapatkan citra dengan kontras yang lebih baik dari sebelumnya. Proses *contrast stretching* dilakukan bergantung pada *graylevel* satu pixel citra dan tidak bergantung pada piksel lain atau bersifat point operations [8]. Point operations dilakukan pada domain spasial, yakni operasi pada sumbu x dan y pada bidang kartesian, atau jika dalam citra disebut baris dan kolom [9].

Proses ini juga disebut dengan pelabelan, pelabelan dan perubahan ukuran citra Pada tahap ini data citra akan mengalami perbaikan kualitas citra. Tahapan pertama setiap gambar akan diberi label untuk menyimpan informasi citra. Selanjutnya citra akan dilakukan perubahan ukuran untuk meningkatkan performa model YOLO.

D. Arsitektur YOLO

Konfigurasi layer YOLO digunakan sebagai jaringan konvolusi untuk melakukan proses training dan normalisasi dataset [10]. Arsitektur konfigurasi jaringan Yolo terbentuk dari layer konvolusi dengan ukuran kernel 3 x 3 dan 1x1 sebagai ekstraksi fitur. Serta lapisan max polling dengan kernel ukuran 2 x 2. Pada bagian akhir, output dataset dikecilkan menjadi 26 x 26 x 18, dimana 26 x 26 adalah ukuran grid dan 18 didapat dari penjumlahan filter.

```
conv     32       3 x 3/ 2    416 x 416 x   3 ->  208 x 208 x  32 0.075 BF
   1 conv     64       3 x 3/ 2    208 x 208 x  32 ->  104 x 104 x  64 0.399 BF
   2 conv     64       3 x 3/ 1    104 x 104 x  64 ->  104 x 104 x  64 0.797 BF
   3 route  2                                1/2 ->  104 x 104 x  32
   4 conv     32       3 x 3/ 1    104 x 104 x  32 ->  104 x 104 x  32 0.199 BF
   5 conv     32       3 x 3/ 1    104 x 104 x  32 ->  104 x 104 x  32 0.199 BF
   6 route  5 4                              ->  104 x 104 x  64
   7 conv     64       1 x 1/ 1    104 x 104 x  64 ->  104 x 104 x  64 0.089 BF
   8 route  2 7                              ->  104 x 104 x 128
   9 max                2x 2/ 2    104 x 104 x 128 ->   52 x  52 x 128 0.001 BF
  10 conv    128       3 x 3/ 1     52 x  52 x 128 ->   52 x  52 x 128 0.797 BF
  11 route  10                               1/2 ->   52 x  52 x  64
  12 conv     64       3 x 3/ 1     52 x  52 x  64 ->   52 x  52 x  64 0.199 BF
  13 conv     64       3 x 3/ 1     52 x  52 x  64 ->   52 x  52 x  64 0.199 BF
  14 route  13 12                           ->   52 x  52 x 128
  15 conv    128       1 x 1/ 1     52 x  52 x 128 ->   52 x  52 x 128 0.089 BF
  16 route  10 15                           ->   52 x  52 x 256
  17 max               2x 2/ 2     52 x  52 x 256 ->   26 x  26 x 256 0.001 BF
  18 conv    256       3 x 3/ 1     26 x  26 x 256 ->   26 x  26 x 256 0.797 BF
  19 route  18                               1/2 ->   26 x  26 x 128
```

Gambar 3. Struktur Layer Konvolusi YOLO

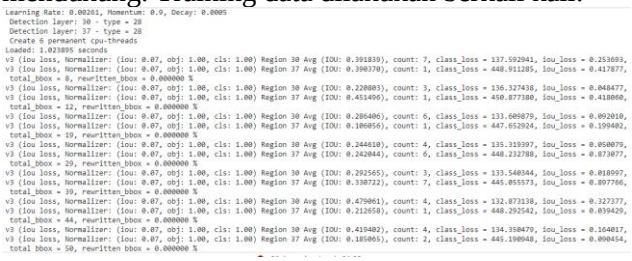
Total keseluruhan layer konvolusi yang digunakan pada training ini berjumlah 36 layer. Kemudian di bagian Filter di tiap [convolutional] layer sebelum [yolo] layer diatur dengan nilai 18 yang diperoleh dari rumus:

Filter = (jumlah kelas dataset+5) × 3 (2)

E. Training Data

Training data dilakukan menggunakan Google Colaboratory atau Google Colab. Google colab merupakan layanan cloud gratis yang disediakan oleh Google dalam bentuk *executable document* yang dapat digunakan untuk menulis, mengedit, menyimpan serta membagikan program yang telah ditulis melalui Google Drive [11]. Proses training data set dilakukan dengan bantuan Google Colab, karena google colab menyediakan GPU dengan kapasitas 12 GB dengan support N-Vidia. Sehingga memungkinkan proses training yang lebih cepat pada

perangkat yang memiliki spesifikasi yang kurang mendukung. Training data dilakukan berkali kali.



Gambar 4. Proses Training Data

F. Pengujian

Pada tahapan pengujian adalah menguji data hasil training dapat mendeteksi citra api. Jika citra api dapat terdeteksi dengan adanya *bounding box*, maka proses training berhasil. Pada proses pengujian spesifikasi PC yang digunakan yaitu seperti pada tabel 1.

Tabel 1. Spesikasi PC yang Digunakan

Processor	AMD Ryzen 5 4600H with Radeon Graphics 3.00 GHz
RAM	16 GB DDR4-3200 SDRAM
Storage	SSD 512 GB PCIe® NVMe™ M.2
Graphic Card	NVIDIA® GeForce® GTX 1650 4GB
Operating System	Windows 10

Selanjutnya adalah melakukan analisis akurasi menggunakan *confusion matrix*. Pengujian dan evaluasi dilakukan untuk mengetahui sensitifitas dan spesifisitas. Pengujian sensitivitas adalah metode perhitungan akurasi dengan membandingkan jumlah klasifikasi tepat pada suatu class dengan keseluruhan klasifikasi yang terdapat pada class tersebut. Sedangkan pengujian spesifisitas adalah metode perhitungan dengan membandingkan jumlah klasifikasi yang tidak berhubungan dengan sebuah class namun dianggap tepat dengan seluruh klasifikasi yang tidak berhubungan dengan class tersebut. Dalam perhitungan selanjutnya akan digunakan *confusion matrix* dalam perhitungan dari beberapa class. Bentuk dari *confusion matrix* dapat dilihat seperti pada tabel 1.

Tabel 1. *Confusion Matrix* Untuk Klasifikasi

		Predicted Class	
		Positive	**Negative**
Actual Class	**Positive**	True Positives (TP)	False Negatives (FN)
	Negative	False Positives (FP)	True Negatives (TN)

Pada tabel 1 terdapat tabel kolom objek yaitu class dari data yang digunakan, serta kolom hasil klasifikasi. Apabila terdeteksi api dan sistem dapat mendeteksi benar api maka akan dikelompokan menjadi *True Positive* (TP) karena hasil klasifikasi sesuai seperti yang diharapkan. Pengelompokan ini didasari oleh hasil deteksi yang menunjukan benar adanya api (true) dan hasil klasfikikasi sesuai dengan yang seharusnya (positive).

Apabila objek bukan api namun sistem mengidentifikasi sebagai api, maka akan dikelompokan menjadi *False Positive* (FP), karena hasil deteksi tidak seperti yang diharapkan. Pengelompokan ini berdasarkan pada objek bukan api (False) namun teridentifikasi sebagai api (Positive). Ketika objek api dideteksi sebagai bukan api, maka akan dikelompokan menjadi *False Negative* (FN). Pengelompokan tersebut didasari dari adanya objek api namun sistem mendeteksi bukan api. Serta apabila objek bukan api terdeteksi sebagai bukan api, maka akan dikelompokan ke *True Negative* (TN). Untuk mendapatkan hasil akurasi maka dari akumulasi tersebut akan dihitung menggunakan rumus sebagai berikut:

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

Untuk menghitung sensitivitas digunakan rumus:

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

Untuk menghitung spesifisitas digunakan rumus:

$$Spesifisitas = \frac{TN}{TN + FP}$$

G. Implementasi Desain Sistem

Tahap implemetasi dari pengembangan yang telah dilakukan yaitu dengan menggunakan kamera webcam yang terhubung melalui kabel USB yang akan menangkap citra secara langsung dan terhubung ke PC untuk diproses.

III. HASIL DAN PEMBAHASAN

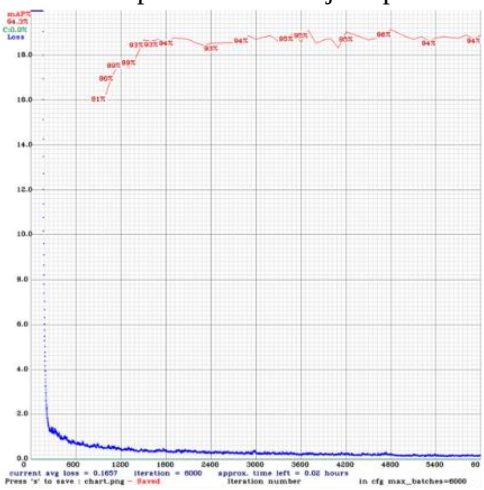
Dalam sistem deteksi, ground station (laptop) harus mampu mendeteksi citra api menggunakan YOLOv4. Tahapan yang dilakukan yaitu mengumpulkan dataset berupa gambar maupun video yang diubah menjadi gambar dengan total 2000 gambar berbagai jenis api, kemudian dilakukan pelabelan.

A. Pembentukan Model Deteksi

Pembentukan model adalah langkah untuk setelah proses training yang ditempuh untuk membentuk model atau weight dari sistem untuk deteksi menggunakan YOLOv4 yang merupakan fungsi untuk untuk mendeteksi objek. Pada prosesnya dalam membentuk model YOLOv4, langkah pertama adalah membentuk model sebelum dilakukan pelatihan atau training yaitu dengan memberi anotasi atau label pada 2000 citra gambar api yang telah

dikumpulkan. Setelah dianotasi maka setiap citra gambar akan memiliki masing masing satu file annotation berformat .xml yang memuat informasi filename, width, height, dan depth. Filename adalah variabel nama daris setiap gambar, width merupakan ukuran panjang citra gambar, height adalah ukuran lebar citra gambar dan depth memuat informasi layer warna dari citra yang digunakan. Kemudian parameter lainnya yang terkandung seperti xmin, ymin, xmax, dan ymax dan parameter lanjutan yaitu letak dari objek yang dimaksud dalam citra gambar tersebut.

Setelah pemberian anotasi pada setiap citra gambar yang merupakan data latih, kemudian tahapan selanjutnya akan dilakukan training atau pelatihan. Pada penelitian ini menggunakan bobot YOLOv4 versi besar atau weight. Proses training akan membentuk file berekstensi weight yang akan digunakan dalam pendeteksian objek api.



Gambar 5. Hasil Training Dataset

Berdasarkan hasil training yang dilakukan untuk membentuk file weight yaitu selama 8 jam dengan iterasi sebanyak 6000 epoch. Pada gambar 5 didapat akurasi terbaik yaitu 96%.

B. Hasil Pengujian

Data hasil training berupa file weight, cfg dan names yang akan digunakan sistem dalam pendeteksian api.

Gambar 6. Pengujian Deteksi Api

Sistem yang telah dibuat dapat dijalankan secara realtime menggunakan kamera webcam, video yang direkam oleh webcam akan langsung diolah untuk mendeteksi api. Pendeteksian menggunakan program yang telah dibuat menggunakan YOLOv4 dalam bahasa python. Pengujian yang telah dilakukan seperti pada gambar 6 dan 7, terbukti sistem mampu mendeteksi api secara realtime dengan pemberian *bounding box* jika positif terindentifikasi sebagai api. Pengujian dilakukan dengan jarak api dengan kamera sebesar 5 meter.

Gambar 7. Pengujian Deteksi Api

C. Evaluasi *Confusion Matrix*

Tabel 2. Hasil Perhitungan *Confusion Matrix*

Video	TP	TN	FP	FN	Recall	Precission	Accuracy
1	46	38	9	1	0,98	0,84	0,89
2	72	36	5	1	0,99	0,94	0,95
3	37	12	5	0	1,00	0,88	0,91
4	77	129	18	1	0,99	0,81	0,92

Precision dan *recall* adalah dua perhitungan yang banyak digunakan untuk mengukur kinerja sistem. Pada penelitian ini precision dihitung untuk mengetahui tingkat ketepatan antara informasi yang diminta dengan jawaban sistem, sedangkan *recall* adalah tingkat keberhasilan sistem dalam menemukan sebuah informasi dan akurasi sebagai tingkat kedekatan antara nilai prediksi dengan nilai aktual.

Dalam pengujian akurasi dilakukan perhitungan menggunakan *confusion matrix*. Nilai confusion matriks yang didapat dari hasil pengujian yang telah dilakukan. Dapat dilihat pada tabel 1, didapat nilai akurasi yang paling baik yaitu dalam pemantauan video 2 dengan durasi 73 detik, sebesar 96%, dengan nilai TP yaitu 73 dan FN hanya 1.

IV. KESIMPULAN

Berdasarkan hasil pengujian yang dilakukan, pengolahan citra menggunakan YOLOv4, sistem dapat mendeteksi api secara realtime, dengan training sebanyak 6000 epoch sehingga mendapatkan akurasi terbaik yaitu sebesar 96%. Serta berdasarkan hasil pengujian dan analisis menggunakan *confusion matrix*, didapat nilai akurasi terbaik yaitu pada video 2 sebesar 0,95 dengan nilai recall yaitu 0,99 dan nilai presisi sebesar 0,95.

UCAPAN TERIMA KASIH

Ucapan terimakasih disampaikan kepada LP2M Universitas Negeri Malang yang telah mendanai penelitian serta semua pihak yang telah mendukung sehingga penelitian ini dapat diimplementasikan dan dituangkan kedalam bentuk tulisan agar bisa menjadi sumber ilmu dan informasi yang bermanfaat.

A. Referensi

[1] Novianto. Luas hutan Indonesia menyusut. Diakses pada https://beritagar.id/artikel/berita/luas-hutan-indonesia-menyusut. 2018.

[2] Rasyid, F. Permasalahan dan Dampak Kebakaran Hutan. Kementerian Lingkungan Hidup dan Kehutanan. 2014.

[3] Pradana, S.Y., Utaminingrum, F., Kurniawan, W., Deteksi Titik Api Terpusat Menggunakan Kamera Dengan Notifikasi Berbasis Sms Gateway Pada Raspberry Pi. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, Vol. 2, No. 12. 2018.

[4] Perkasa, P., Aguswan, Y., Use of Drone for Early Detection Means Land and Forest Fire. Jurnal Pendidikan Teknologi dan Kejuruan BALANGA. 2018.

[5] Gularso, H., Subiyanto S., Sabri L.M., Tinjauan Pemotretan Udara Format Kecil Menggunakan Pesawat ModelSkywalker 1680 (Studi Kasus: Area Sekitar Kampus UNDIP). Jurnal Geodesi UNDIP. 2013

[6] Firdausy, K., Saudi, Y., Sutikno, T., Deteksi Api Real-Time Dengan Metode Thresholding Rerata RGB. Jurnal TELKOMNIKA. Vol. 5, No. 2. 2007.

[7] Bochkovskiy, A., Wang, C., Liao H.M., YOLOv4: Optimal Speed and Accuracy of Object Detection. SIJ 50: DIGILIENCE 2021: AI-driven Cybersecurity Solutions, Cyber Ranges, and Military ICT Applications. 2020

[8] Supiyatno, Suparwati, T., Perbaikan Citra Menggunakan Metode Contrast Stretching. Jurnal Siger Matematika. Vol. 2, No. 1. 2021.

[9] Yudistiawan, Implementasi Metode Contrast Stretching Untuk Penajaman Citra Digital. Jurnal Buffer Informatika. Volume 4 Nomor 2. 2018.

[10] Pamungkas, B.P.G., Nugroho, B., Anggraeny, F. Deteksi Dan Menghitung Manusia Menggunakan Yolo-CNN. Jurnal Informatika dan Sistem Informasi (JIFoSI). 2021

[11] Sengkey, D.F., Kambey, F.D., Lengkong, S.P., Joshua, S.R., Kainde, H.V.F., Pemanfaatan Platform Pemrograman Daring dalam Pembelajaran Probabilitas dan Statistika di Masa Pandemi CoVID-19. Jurnal Informatika vol. 15. hal. 257-264. 2020