Logo

Description automatically generated

**Rancang Bangun Sistem Deteksi Api pada Sistem Tertanam menggunakan Algoritma YoloV4 berbasis IoT**

**PROPOSAL SKIRIPSI**

**ANNASTYA BAGAS DEWANTARA**

**1910314024**

**PROGRAM STUDI TEKNIK ELEKTRO**

**FAKULTAS TEKNIK**

**UNIVERSITAS PEMBANGUNAN NASIONAL VETERAN JAKARTA**

**2022**

# HALAMAN PERNYATAAN

Proposal skripsi ini merupakan hasil karya sendiri, dan semua sumber yang dikutip maupun dirujuk telah saya nyatakan benar.

Nama : Annastya Bagas Dewantara

NIM : 1910314024

Program Studi : Teknik Elektro

Bilamana dikemudian hari ditemukan ketidaksesuaian dengan pernyataan saya ini, maka saya bersedia dituntut dan diproses sesuai dengan ketentuan yang berlaku.

Jakarta, 05 Desember 2022

Yang menyatakan,

(Annastya Bagas Dewantara)

# LEMBAR PENGESAHAN

**Rancang Bangun Sistem Deteksi Api pada Sistem Tertanam menggunakan Algoritma YoloV4 berbasis IoT**

**Annastya Bagas Dewantara**

**NIM 1910314024**

**Disetujui Oleh**

|  |  |
| --- | --- |
| **Pembimbing I** | **Pembimbing II** |
|  |  |
| **Achmad Zuchriadi S.T., M,T NIP.219111150** | **Fajar Rahayu S.T., M.T.**  **NIP.217121308** |

**Mengetahui,**

**Ketua Program Studi Teknik Elektro**

**Fakultas Teknik Universitas Pembangunan Nasional Veteran Jakarta**

**Achmad Zuchriadi S.T., M,T**

**NIP.219111150**

# KATA PENGANTAR

Puji syukur kepada Allah SWT atas kehendak-Nya penulis dapat menyusun proposal skripsi dengan baik dan tidak terkendala apa pun. Judul yang penulis pilih dalam penelitian ini adalah “Rancang Bangun Sistem Deteksi Api pada Sistem Tertanam menggunakan Algoritma YoloV4 berbasis IoT. Dalam pembuatan proposal skripsi ini, tidak sedikit masalah yang harus dihadapi oleh penulis. Namun penulis menyadari bahwa kelancaran dalam penyusunan laporan akhir ini tidak luput dari bantuan, dorongan dan bimbingan banyak pihak, sehingga masalah yang dihadapi penulis dapat teratasi dan menyelesaikan laporan ini dengan tepat waktu. Oleh karena itu penulis ingin mengucapkan terima kasih kepada :

1. Allah SWT. yang telah memberikan rahmat dan kemudahan bagi penulis dalam menyelesaikan proposal skripsi.
2. Ibu Fajar Rahayu S.T.,M.T selaku dosen pembimbing I skripsi yang telah memberikan dukungan serta banyak saran yang sangat bermanfaat.
3. Bapak Achmad Zuchriadi S.T.,M.T selaku dosen pembimbing II yang telah banyak memberikan saran serta masukkan yang sangat bermanfaat.
4. Keluarga yang selalu memberikan dorongan kepada penulis agar dapat menyelesaikan skripsi ini.
5. Teman-teman Program Studi S1 Teknik Elektro angkatan 2019 Fakultas Teknik yang telah memberikan semangat, dukungan dan doa sehingga skripsi ini dapat selesai tepat waktu.

Jakarta, 05 Desember 2022

Penulis,

Annastya Bagas Dewantara

# daftar isi

[HALAMAN PERNYATAAN i](#_Toc121148248)

[LEMBAR PENGESAHAN ii](#_Toc121148249)

[KATA PENGANTAR iii](#_Toc121148250)

[DAFTAR ISI iv](#_Toc121148251)

[DAFTAR GAMBAR vi](#_Toc121148252)

[DAFTAR TABEL vii](#_Toc121148253)

[BAB 1](#_Toc121148254) [PENDAHULUAN 1](#_Toc121148255)

[1.1 Latar Belakang 1](#_Toc121148256)

[1.2 Tujuan Penelitian 2](#_Toc121148257)

[1.3 Rumusan Masalah 2](#_Toc121148258)

[1.4 Batasan Masalah 2](#_Toc121148259)

[1.5 Sistematika Penulisan 3](#_Toc121148260)

[BAB 2](#_Toc121148261) [LANDASAN TEORI 4](#_Toc121148262)

[2.1 State of Art 4](#_Toc121148263)

[*2.2* *Convolutional Neural Network (CNN)* 6](#_Toc121148264)

[2.3 YoloV4 8](#_Toc121148265)

[2.4 Raspberry Pi 15](#_Toc121148267)

[BAB 3](#_Toc121148268) [METODOLOGI PENELITIAN 17](#_Toc121148269)

[3.1 Tahapan Penelitian 17](#_Toc121148270)

[3.2 Implementasi 18](#_Toc121148271)

[3.3 Implementasi 19](#_Toc121148272)

[REFERENSI 20](#_Toc121148273)

# daftar gambar

[Gambar 3.1 Arsitektur Convolutional Neural Network 7](#_Toc121148625)

[Gambar 3.2 Lapisan Konvolusi 7](#_Toc121148626)

[Gambar 3.3 Lapisan *Max Pooling* 8](#_Toc121148627)

[Gambar 3.4 Lapisan *Fully Connected* 8](#_Toc121148629)

[Gambar 3.5 Visualisasi *Intersection of Union* (IoU) 9](#_Toc121148630)

[Gambar 3.6 *Non-max Suppression* (NMS) 10](#_Toc121148631)

[*Gambar 3.7 Object Localization* dan *Classification* 10](#_Toc121148632)

[Gambar 3.8 *Residual Block* pada sampel gambar 11](#_Toc121148633)

[Gambar 3.9 Tahapan deteksi menggunakan Algoritma YOLO 13](#_Toc121148634)

[*Gambar 3.10* Nilai interpolasi AP berdasarkan nilai maksimum antara *Recall-Precission* 15](#_Toc121148635)

[*Gambar 3.11 mean Average-Precission* (mAP) 15](#_Toc121148636)

[Gambar 4.1 Diagram Alir Penelitian 17](#_Toc121148637)

[Gambar 4.2 Diagram Alur Kerja Alat 18](#_Toc121148638)

# daftar tabel

[Tabel 2.1 State of Art Penelitian 4](#_Toc121148679)

[Tabel 3.1 Spesifikasi Perangkat *Training* 18](#_Toc121148680)

[Tabel 3.2 Spesifikasi Perangkat Interferensi 19](#_Toc121148681)

[Tabel 3.3 Jadwal Penelitian 19](#_Toc121148682)

# 

# PENDAHULUAN

## Latar Belakang

Pertumbuhan jumlah populasi di dunia kian meningkat tiap tahunnya, dengan angka sebesar 83 juta per tahun atau sebesar 1.1%. Peningkatan populasi global telah tumbuh dari 1 miliyar pada tahun 1800 ke angka 7.9 miliyar di tahun 2020 dan telah di prediksi akan mencapai angka 8.6 miliyar pada pertengahan tahun 2030 dan 9.8 miliyar di tahun 2050 [1] Pertumbuhan dan peningkatan pertumbuhan penduduk akan berbanding lurus dengan peningkatan kepadatan penduduk khususnya di daerah perkotaan. Tingginya aktivitas perkotaan yang diiringi oleh kurangnya perencanaan dan penyediaan lahan pemukiman yang layak mengakibatkan tingkat potensi kebakaran meningkat [2].

Kebakaran pada daerah dalam gedung seperti perumahan, perkantoran dan pemukiman merupakan penyebab tertinggi terjadinya kebakaran, dengan penyebab terbesar pada memasak, malfungsi listrik dan merokok [3]. Tingginya angka kecelakaan dan kematian yang dialami oleh pemadam api dalam usaha memadamkan api akibat terjatuh, tabrakan kendaraan serta luka bakar akibat panas api dan ledakan [4], menjadi landasan pada peningkatan tindakan pencegahan kebakaran. Pada penerapan pencegahan kebakaran menggunakan perangkat deteksi asap konvensional terdapat keterbatasan dalam mengukur intensitas tingkat kebakaran dan jangkauan terbatas dalam mendeteksi kebakaran. Menggunakan metode *Artificial Intelligence* dan *Machine Learning* yang di tambahkan sensor infrared untuk mendeteksi panas, objek api dapat dideteksi dengan jangkauan yang lebih luas dan akurasi yang lebih tinggi.

Beberapa metode yang telah digunakan dalam studi literatur pada deteksi objek api pada *embedded system,* Deteksi objek api dengan menggunakan *Haar Cascade Classifier* pada YoloV3 [5], Deteksi api pada *noisy image* menggunakan *Faster R-CNN* model [6], *Bayes Classifier* dan *Support Vector Machine* (SVM) pada gambar RGB [7] Algoritma *Single Shot Multibox Detector* (SSD) pada UAV [8] dan penerapan deteksi kebakaran hutan menggunakan EfficientDet-Lite dan Yolov5 [9].

Kebakaran dapat terjadi kapan pun tanpa disadari. Sistem deteksi api yang dapat mengirimkn alarm dan notifikasi kepada pemiliknya akan sangat berguna terutama di beberapa situasi. Sistem inimenggunakan deteksi objek dengan algoritma YoloV4 pada sistem tertanam Raspberry Pi 4 yang tersambung dengan kamera. Lokalisasi objek dilakukan untuk mendeteksi lokasi piksel dari frame, sistem mampu mendeteksi lokasi terjadinya api. Sistem deteksi api mampu menyalakan alarm melalui modul suara yang tertanam padanya, dan mampu mengirimkan notifikasi kepada pengguna melalui sistem IoT via *HTML request* untuk melakukan *monitoring* secara ­*real-time.*

## Tujuan Penelitian

### Merancang desain sistem monitoring dari Sistem Deteksi Api.

### Merancang desain sirkuit alert dari Sistem Deteksi Api.

### Merancang model deteksi objek dari Sistem Deteksi Api.

### Melakukan analisis dan evaluasi terhadap hasil *recall, precission* dan *accuracy* dari model yang telah dibuat.

## Rumusan Masalah

### Bagaimana desain sistem *monitoring* dari Sistem Deteksi Api?

### Bagaimana desain *sistem* alert dari Sistem Deteksi Api *t*?

### Bagaimana model deteksi objek dari Sistem Deteksi Api?

### Bagaimana hasil analisis dan evaluasi terhadap hasil *recall, precission* dan *accuracy* dari model yang telah dibuat?

## Batasan Masalah

### Klasifikasi dibagi ke dalam tiga kelas yakni api menyala, berasap dan padam.

### Sistem *monitoring* dilakukan pada server lokal berbasis web.

## Sistematika Penulisan

Penelitian ini disusun dan diuraikan dengan sistematika sebagai berikut:

Bab I Pendahuluan

Bab ini berisi latar belakang, rumusan masalah, tujuan, batasan permasalahan, dan sistematika penulisan.

Bab II Tinjauan Pustaka

Bab ini berisikan dasar ilmu yang mendukung dan penelitian terdahulu yang berkaitan dengan penelitian ini.

Bab III Metodologi Penelitian

Bab ini berisi rancangan penelitian berupa metode penelitian digunakan, prosedur penelitian berupa flowchart, dimana dalam tahap ini akan dilakukan pengolahan citra digital sesuai dengan metode dan algoritma yang telah ditetapkan. Penelitian ini menggunakan metode eksperimental dalam melakukan uji coba algoritma Yolov4 pada *embedded system* dalam deteksi api.

Bab IV Hasil dan Pembahasan

Bab ini berisi proses uji coba berdasarkan parameter – parameter yang ditetapkan, dan kemudian dilakukan analisa terhadap hasil uji coba tersebut.

Bab V Penutup

Bab ini berisi kesimpulan yang dapat diambil dari penelitian ini serta saran untuk pengembangan selanjutnya.

# 

# landasan teori

## State of Art

Penelitian ini dilakukan dengan memperhatikan beberapa penelitian terdahulu yang membahas mengenai tema yang bersinggungan terhadap Sistem Deteksi Api meliputi sistem IoT, penerapan pada sistem tertanam, deteksi objek, dan sistem monitoring . Beberapa penelitian terdahulu dapat dilihat pada tabel 2.1.1 dibawah ini:

#### State of Art Penelitian

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Judul Jurnal** | **Tipe** | **Ringkasan** |
| 1 | M. Kanwar and L. Agilandeeswari, "IOT Based Fire Fighting Robot," 2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), 2018, pp. 718-723, doi: 10.1109/ICRITO.2018.8748619. [10] | IoT | Sistem monitoring dan kontrol jarak jauh pada robot pemadam api berbasis IoT pada Raspberry Pi dan Arduino yang dihubungkan dengan sensor ultrasonik dan sensor deteksi api. |
| 2 | S. Zhao, B. Liu, Z. Chi, T. Li and S. Li, "Characteristics Based Fire Detection System Under the Effect of Electric Fields With Improved Yolo-v4 and ViBe," in IEEE Access, vol. 10, pp. 81899-81909, 2022, doi: 10.1109/ACCESS.2022.3190867. [11] | Deteksi Api dengan YoloV4 | Mendeteksi api yang muncul akibat penerapan perlakuan efek medan elektrik yang berbeda menggunakan YoloV4. |
| 3 | Kulshreshtha M, Chandra SS, Randhawa P, Tsaramirsis G, Khadidos A, Khadidos AO. OATCR: Outdoor Autonomous Trash-Collecting Robot Design Using YOLOv4-Tiny. Electronics. 2021; 10(18):2292.https://doi.org/10.3390/electronics10182292. [12] | Yolov4 pada Sistem Tertanam | Navigasi kontrol robot menggunakan DC motor yang terhubung *driver* motor L298N serta *stepper* motor berdasarkan koordinat pixel dari objek sampah pada YoloV4. |
| 4 | Madhar, Madhar. "Rancang Bangun Sistem Monitoring Deteksi Dini Kebakaran Dengan Fitur Gps Berbasis Website." JATI (Jurnal Mahasiswa Teknik Informatika) 2.1 (2018): 367-372. [13] | Sistem Monitoring Web | Sistem Monitoring web menggunakan ESP8266 menggunakan LM35, DHT11 dan GPS dalam memberikan informasi lokasi dan suhu dalam mendeterminasikan tingkat kebakaran. |
| 5 | Suwarjono, Suwarjono, Izak Habel Wayangkau, Teddy Istanto, Rachmat Rachmat, Marsujitullah Marsujitullah, Hariyanto Hariyanto, Wahyu Caesarendra, Stanislaw Legutko, and Adam Glowacz. 2021. "Design of a Home Fire Detection System Using Arduino and SMS Gateway" Knowledge 1, no. 1: 61-74. https://doi.org/10.3390/knowledge1010007. [14] | Sistem *Alerting* | Sistem *alerting* yang terhubung dengan sensor api dan Arduino UNO. Intensitas pembacaan pada sensor api apabila melebihi batasan tertentu maka *buzzer* akan berbunyi dan mengirimkan pesan peringatan melalui omodul GSM. |

Keterbaharuan yang terdapat pada penelitian ini terletak pada pengimplementasian algoritma YoloV4 dalam mendeteksi api ke dalam perangkat sistem tertanam yang terintegrasi IoT.

## *Convolutional Neural Network (CNN)*

*Deep Learning* adalah ilmu di bidang *Machine Learning* yang berkembang karena perkembangan GPU teknologi akselerasi yang memiliki kemampuan luar biasa dalam *computer vision*. Salah satu kemampuannya ada di kasus klasifikasi objek dalam gambar. Untuk menerapkan Deep Learning yang dapat digunakan untuk objek gambar klasifikasi, yaitu CNN. Cara kerja CNN mirip dengan *Multi Layer Perceptron*  (MLP), tetapi di CNN, setiap neuron disajikan dalam dua dimensi. MLP menerima input data pada satu dimensi dan meneruskan informasi ke keluaran *Neural Network*. Setiap tautan antara neuron dalam dua lapisan yang berdekatan memiliki parameter bobot satu dimensi yang menentukan kualitas mode. Pada setiap lapisan data input, operasi linier dilakukan dengan nilai bobot yang ada, kemudian hasil komputasi tersebut ditransformasikan menggunakan operasi non-linier yang disebut fungsi aktivasi. Di CNN, data yang disebarkan pada jaringan adalah dua dimensi, sehingga operasi linier dan parameter bobot pada CNN berbeda. Di CNN, operasi linier menggunakan operasi konvolusi, sedangkan beratnya tidak lagi satu dimensi tetapi empat dimensi yang merupakan kumpulan dari konvolusi kernel. Sebuah CNN terdiri dari beberapa jenis layer, yaitu *Convolutional Layer, Max Pooling Layer*, dan *Fully Connected Layer* [15]*.*

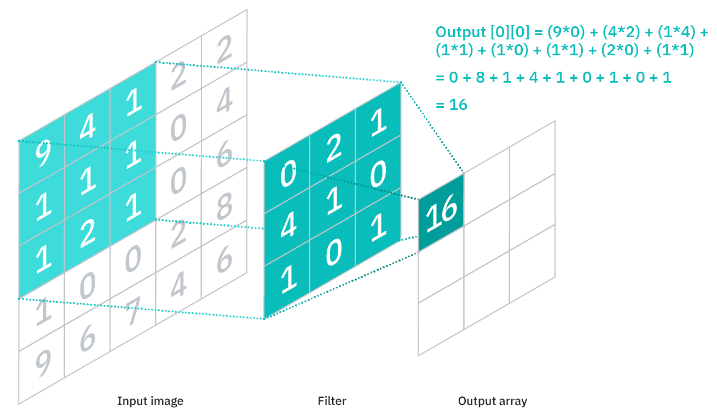
Diagram

Description automatically generated

##### Arsitektur Convolutional Neural Network

### *Convolutional Layer*

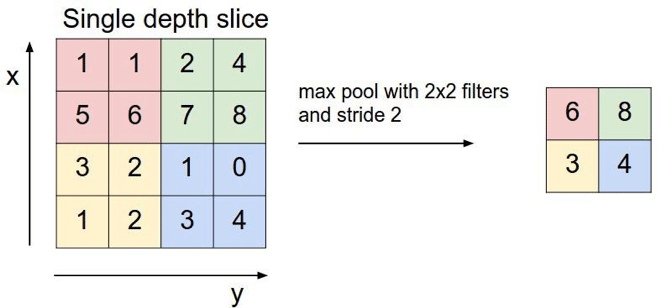
Lapisan konvolusi adalah blok bangunan utama CNN. Ini berisi satu set filter (atau kernel). Ukuran filter biasanya lebih kecil dari gambar sebenarnya. Setiap filter menyatu dengan gambar dan membuat peta aktivasi. Untuk konvolusi, filter meluncur melintasi tinggi dan lebar gambar dan *dot product* antara setiap elemen filter dan input dihitung pada setiap posisi spasial [16].



##### Lapisan Konvolusi

### *Max Pooling Layer*

Fungsi dari pooling ini adalah untuk mereduksi input secara spasial (mengurangi jumlah parameter) dengan operasi down-sampling. Umumnya, metode pooling yang digunakan adalah max pooling atau mengambil nilai terbesar dari bagian tersebut. Namun terdapat metode pooling lain yang dapat digunakan seperti average pooling atau L2-norm pooling.



##### Lapisan *Max Pooling*

### *Fully Connected Layer*

Neuron yang telah melewati lapisan *Convolution* dan *Max Pooling Average* diteruskan ke lapisan *Flatten* dimana setiap bagian diubah menjadi vektor sehingga data dapat diklasifikasikan secara

linear. Output diubah menjadi 1 x 1 dimensi dimana jumlahnya didapat dari perkalian antara lebar (width), tinggi (height) dan channel ditambah jumlah bias. Proses ini akan berlanjut pada fungsi aktivasi yang akan menghitung probabilitas dari setiap kelas yang memungkinkan menentukan class dari input citra yang diberikan.

##### 

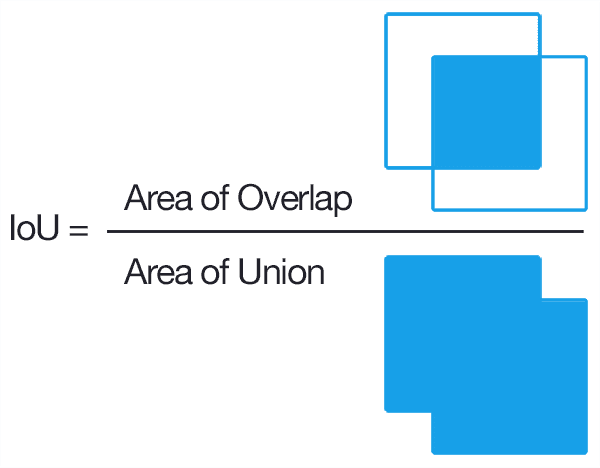
##### Lapisan *Fully Connected*

## YoloV4

YOLO merupakan abreviasi dari *You Only Look Once* adalah algoritma yang digunakan untuk objek deteksi karena ukurannya yang kecil dan kecepatan pada kemampuan deteksinya. YOLO menggunakan *Convolution Neural Network* (CNN) pada *single forward propagation*

### *Intersection over Union (IoU)*

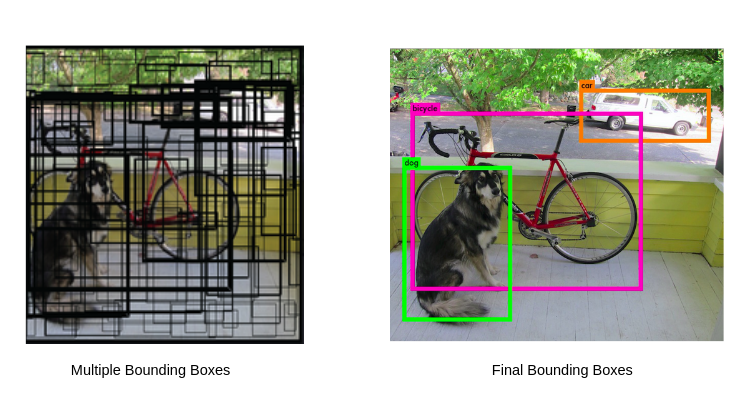
IoU, juga dikenal sebagai indeks Jaccard, adalah yang metrik yang paling umum untuk membandingkan kesamaan antara dua bentuk arbitrer. IoU mengkodekan properti bentuk dari objek yang dibandingkan, yakni lebar, tinggi, dan lokasi dua *bounding boxes*, ke dalam properti wilayah dan kemudian menghitung ukuran yang dinormalisasi yang berfokus pada area gambar[17].



##### Visualisasi *Intersection of Union* (IoU)

Nilai *confidence* didapat dari perhitungan IoU. Pada Computer Vision digunakan untuk mendeteksi objek dengan benar. Saat pelatihan memprediksi objek dengan *bounding box*, maka *bounding box* yang diprediksi dibandingkan dengan bounding box dari *ground-truth.*

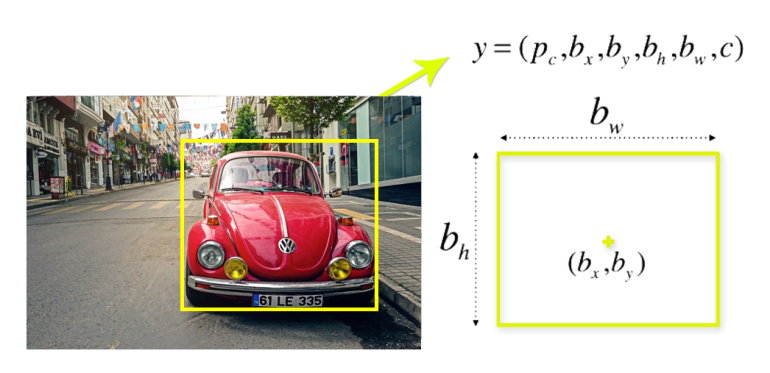
Pada proses deteksi objek dalam gambar, *bounding boxes* seringkali terbentuk. Algoritma *Non-max suppression* (NMS)digunakan untuk mengeliminasi *bounding boxes* terbaik dari gambar. NMS merupakan algoritma yang bertugas untuk mencari local maxima berdasarkan nilai batasan IoU awal model [18] Apabila nilai batasan awal IoU adalah 0.5, maka proses eliminasi pada *bounding boxes* yang memiliki nilai dibawah 0.5 akan di eliminasi, kemudian setelah dilakukan proses eliminasi pertama akan dialnjutkan dengan proses eliminasi kedua dengan mengambil *bounding boxes* dengan *confidence score* terbesar dari *bounding boxes.*



##### *Non-max Suppression* (NMS)

### *Object Localization*

Model lokalisasi objek mirip dengan model klasifikasi. Tetapi model lokalisasi yang terlatih juga memprediksi lokasi objek dalam gambar dengan menggambar kotak pembatas di sekitarnya. Sebagai contoh, sebuah mobil terletak pada gambar di bawah ini. Informasi kotak pembatas, koordinat titik pusat, lebar dan, tinggi juga disertakan dalam keluaran model.



##### *Object Localization* dan *Classification*

*Output* dari lokalisasi model akan mencakup informasi dari bounding boxes, sehingga *output* akan terlihat pada vektor dibawah ini:

|  |  |
| --- | --- |
|  | : *Confidence score* dari objek pada *bounding boxes*  : Pusat koordinat x terhadap bagian kiri *bounding boxes*  : Pusat koordinat y terhadap bagian atas *bounding boxes*  : Tinggi *bounding boxes*  : Lebar *bounding boxes*  : Bernilai satu apabila terdapat kelas-n yang terdeteksi, nol apabila tidak terdapat kelas-n yang terdeteksi\ |

### Grid Cells

YOLO membagi masukan gambar ke dalam  *S*×*S grid.* Apabila pusat dari objek terletak pada *grid cell* maka *grid cell* trsebut bertanggung jawab dalam mendeteksi objek tersebut.Setiap *grid cell* melakukan prediksi terhadap nilai *B Bounding Boxes* dan menghitung *Confidence Score* dari masing-masing *grid cells.*

# 

##### *Residual Block* pada sampel gambar

*Confidene score* merefleksikan tingkat akurasi dan kepercayaan dari model dalam memprediksi objek di dalam *Boxes* [19]*.* Apabila tidak terdeteksi objek pada *cell* maka nilai *Confidence Score* bernilai nol, namun apabila tidak maka nilai *Confidence Score* akan berbanding lurus dengan *bounding boxes* terhadap *ground truth*  (IoU).Persamaan dari nilai *Confidence Score* dapat dihitung dengan persamaan:

: Probabilitas dari *bounding boxes* terdapat objek

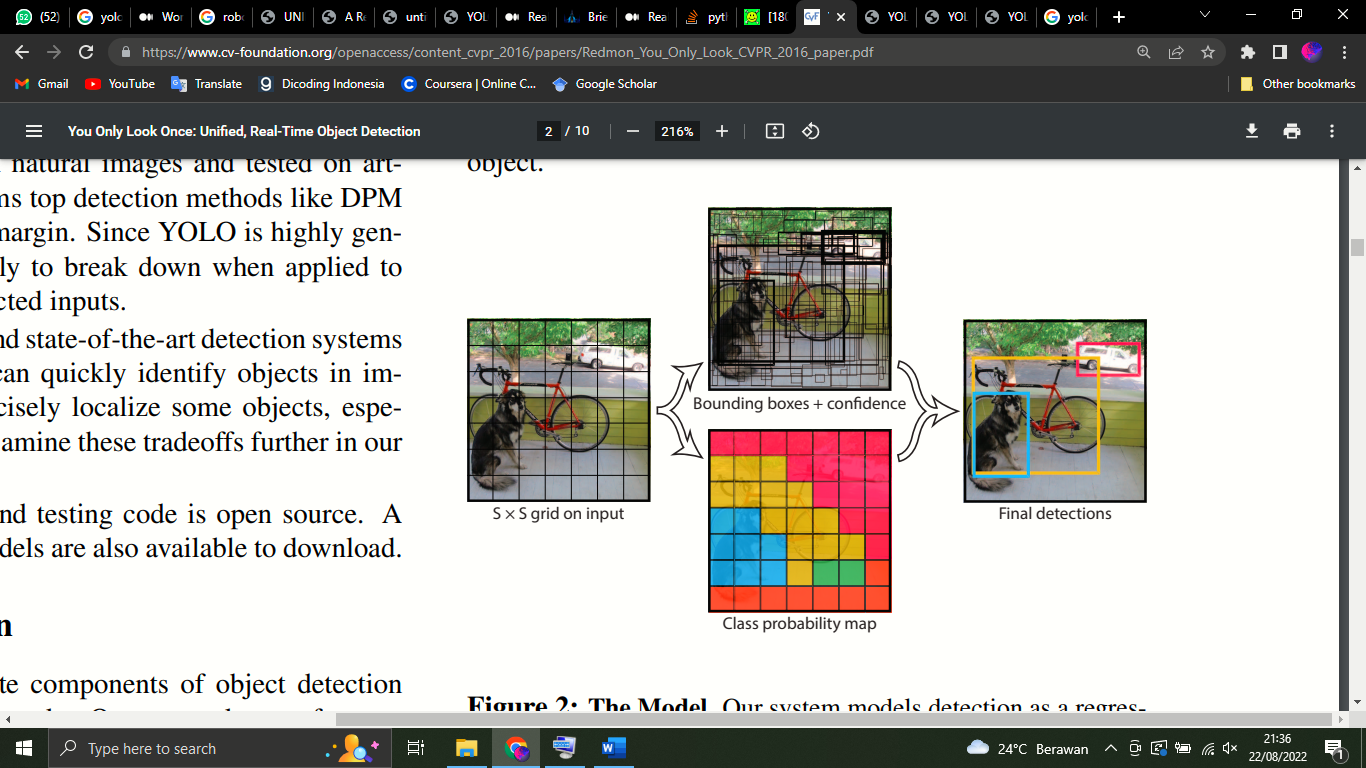
: *Intersection of Union* (IoU)

Setiap *grid cells* memprediksi nilai kondisional dari probabilitas masing-masing kelas. *Grid cell* menghasilkan prediksi kemungkinan kelas terbesar yang kemudian digunakan untuk memprediksi tingkat akurasi kelas berdasarkan nilai IoU nya.

: Probabilitas suatu objek berada di ketika suatu objek terdeteksi

: *Intersection of Union* (IoU)

: Probabitas objek itu berada di



##### Tahapan deteksi menggunakan Algoritma YOLO

### *Data Augmentation*

Data augmentasi merupakan metode yang digunakan untuk meningkatkan masukan input data dengan membuat data replika yang telah di manipulasi untuk mengatasi permasalahan ketidakseimbangan masukan data pada kelas, meningkatkan generalisasi dan mencegah terjadinya *overfitting* [20] YOLOv4 menggunakan beberapa teknik dari data augmentasi yang disebut sebagai “the bag of freebies” karena kemampuannya yang mampu meningkatkan akurasi tanpa memberatkan proses inferensi, adapun augmentasi data yang digunakan pada YOLOV4 yakni *Photometric Distortion, Image Occlusion* dan *Geometric Distortion* [21]

*Photometric Distortion* mencakup augmentasi gambar dengan memanipulasi tingkat saturasi, kecerahan, kontras pada gambar. *Geometric Distortion* mencakup *cropping, flipping* dan *rotating. Image Occlusion* juga digunakan sebagai legularisasi dalam menghindari terjadinya *overfitting* dengan menggunakan *random erase, cut out, hide and seek, CutMix, grid mask* dan *MixUp*.

### *Recall, Precission* dan *mAP*

*Recall* merupakan perbandingan atau proporsi antara kasus *real positive* terhadap *predicted positive. Predicted positive* merupakan penjumlahan antara *true positive* dan *false negative*.

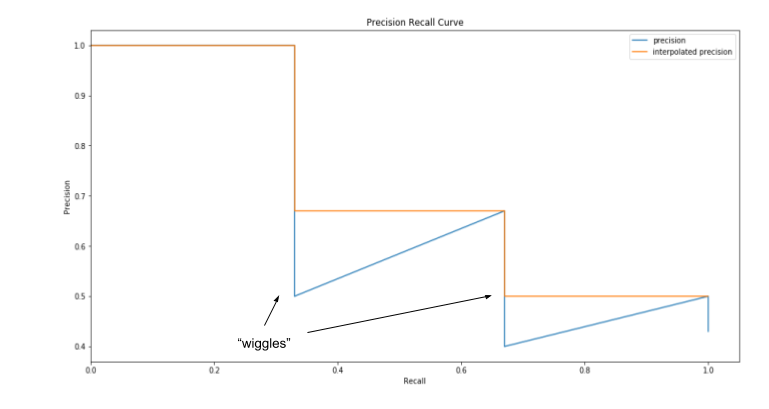
*Precission* merupakan perbandingan atau proporsi antara kasus *predicted positive* yang merupakan *real positive. Predicted positive* merupakan penjumlahan antara *true positive* dan *false negative* [22].

*Average precission* (AP) merupakan area yang terbentuk antara *recall* dan *precission. Average precission* (AP) merupakan standard ukuran yang digunakan untuk mengukur performansi dari objek deteksi. Nilai dari AP digdapatkan berdasarkan nilai awal IoU. Persamaan yang digunakan dalam mencari AP adalah persamaan yang digunakan untuk mencari area antara grafik *Recall-Precission* yakni :

|  |  |
| --- | --- |
|  | Fungsi *precission* terhadap *recall*  : Nilai *recall* |

Proses interpolasi dilakukan terhadap nilai PR untuk memnimalisir variasi yang ditimbulkan akibat urutan sampel. Interpolasi AP dilakukan dengan mengganti nilai *recall* dengan maksimum *precission* pada *recall*

|  |
| --- |
|  |
| : *Precission* yang telah terinterpolasi  : Nilai *recall* dengan *precission* terbesar |



##### Nilai interpolasi AP berdasarkan nilai maksimum antara *Recall-Precission*

Nilai *recall* di bagi ke dalam 11 titik dari interval [0, 1] terhadap nilai AP yang telah dilakukan interpolasi untuk dihitung nilai rata-rata. Nilai rata-rata dari nilai ke-n *recall values* setelah dilakukan interpolasi terhadap nilai maximum *precission* dapat dirumuskan ke dalam persamaan berikut [24] :

|  |
| --- |
|  |
| N : Nilai *recall* pada interval [0, 1]  mAP : *mean Average-Precission*  : Nilai *precission* yang telah ter-interpolasi |

Chart, line chart

Description automatically generated

##### *mean Average-Precission* (mAP)

## Raspberry Pi

Raspberry Pi, sering juga disingkat dengan nama Raspi, adalah komputer papan tunggal yang memiliki ukuran sebesar kartu kredit. Raspberry Pi bisa digunakan untuk berbagai keperluan, seperti spreadsheet, game, bahkan bisa digunakan sebagai media player karena kemampuannya dalam memutar video high definition. Raspberry Pi dikembangkan oleh yayasan nirlaba, Rasberry PiFoundation yang digawangi sejumlah developer dan ahli komputer dari Universitas Cambridge, Inggris.

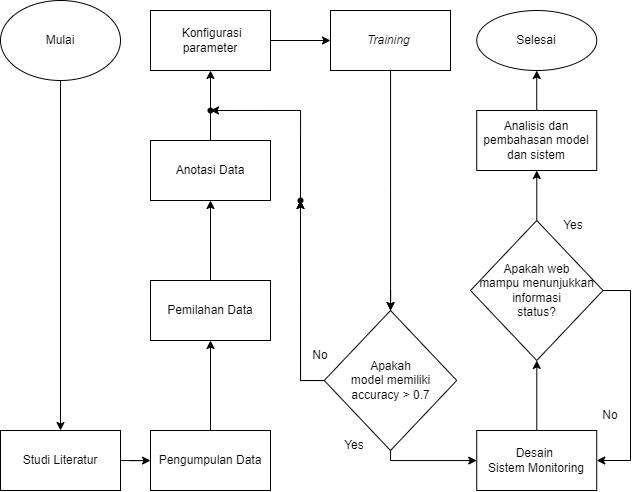
Raspberry Pi memiliki dua model yaitu model A dan model B. Secara umum Raspberry Pi Model B, 512MB RAM. Perbedaan model A dan B terletak pada memori yang digunakan, Model A menggunakan memori 256 MB dan model B 512 MB. Selain itu model B juga sudah dilengkapai dengan ethernet port (kartu jaringan) yang tidak terdapat di model A. Desain Raspberry Pi didasarkan seputar *SoC (System-on-a-chip)* Broadcom BCM2835, yang telah menanamkan prosesor ARM1176JZF-S dengan 700 MHz, VideoCore IV GPU, dan 256 MegabyteRAM (model B). Penyimpanan data didisain tidak untuk menggunakan hard disk atau solidstate drive, melainkan mengandalkan kartu SD (SD memory card) untuk booting dan penyimpanan jangka panjang.

# 

# metodologi penelitian

## Tahapan Penelitian

Tahapan penelitian ini dibagi ke dalam beberapa bagian besar yakni *pre-research* tahapan sebelum penelitian di lakukan dengan melakukan studi literatur terkait penelitian serupa yang memiliki keterkaitan topik yang sedang dibahas. Tahapan selanjutnya adalah *pre-processing* data, yakni tahapan pengolahan data sebelum di *training,* tujuannya untuk meningkatkan akurasi dari prediksi pada proses interferensi. Setelah fase *pre-processing* data dilanjutkan ke fase *Training* dengan menyesuaikan dan mengkonfigurasikan parameter yang digunakan. Fase berikutnya dilanjutkan dengan evaluasi model untuk mengevaluasi model terhadap kriteria yang di tetapkan, apabila model telah memenuhi kriteria yang ditentukan dilakukan analisis dan pembahasan terkait kinerja model pada system deteksi api.



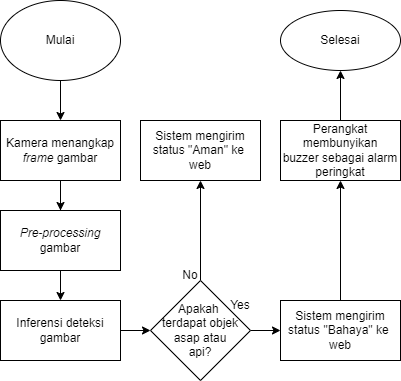
##### Diagram Alir Penelitian

Penelitian di awali dengan melakukn studi literatur terkait proses *pre-processing* gambar dengan menerapkan morfologi, sebagai proses filter dan juga untuk memperkuat fitur-fitur pada gambar. Gambar tersebut kemudian di anotasi sebagai bahan *training* dengan mengkonfigurasi parameter seperti besar *epochs* dan *batch* pada *training.* Informasi prediksi gambar meliputi kelas, skor prediksi dan kondisi akan dikirimkn dan di visualisasikan ke dalam bentuk web. Analisis kinerja dari web dan sistem deteksi di lakukan sebagai optimalisasi dari kinerja sistem deteksi api.

## Implementasi

### Alur Kerja

Proses kerja dari alat di awali dengan perangkat mengambil gambar *frame* per *frame* gambar. *Frame* gambar tersebut di lakukan *pre-processing* dengan menerapkan morfologi sebelum di inferensikan untuk mendapatkan prediksi kelas dan besar skor prediksi. Skor prediksi dan prediksi kelas dari gambar dikirimkan ke *website* untuk di tampilkan kepada pengguna.



##### Diagram Alur Kerja Alat

### Perangkat

Adapun perangkat keras yang digunakan pada fase *Training* dan interferensi :

### *Training*

#### Spesifikasi Perangkat *Training*

|  |  |
| --- | --- |
| GPU : | Nvidia RTX3090Ti |
| Processor : | Intel i7 9750H |
| RAM : | 12 GB |
| Penyimpanan : | SSD *PCIe* 1 TB |

1. Interferensi

#### Spesifikasi Perangkat Interferensi

|  |  |
| --- | --- |
| Model : | Raspberry Pi 4 |
| Processor : | Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz |
| RAM : | 4 GB |
| Penyimpanan : | 32 GB SD Card |

## Implementasi

#### Jadwal Penelitian



# hasil

## Pengambilan Data

Data yang digunakan untuk model objek lokalisasi YOLOv4 pada sistem deteksi kebakaran harus mencakup berbagai jenis kebakaran dengan berbagai tingkat intensitas. Data ini harus diperoleh dari berbagai sumber seperti video kebakaran yang diperolah dari CCTV, rekaman video simulasi kebakaran dan gambar kebakaran.

|  |  |  |
| --- | --- | --- |
|  | **A picture containing green, light, railroad  Description automatically generated** | **A picture containing text, sky, outdoor, green  Description automatically generated** |

##### Pembagian Dataset Api and Asap

Untuk mendapatkan data yang representatif, pengambilan data harus dilakukan di berbagai situasi dan kondisi yang berbeda, data yang di ambil mencakup data api dan asap dengan ukuran daan intensitas yang berbeda. Data yang beragam ini akan membantu model untuk menangani berbagai kondisi yang berbeda dan menghindari terjadinya *overfitting* [12]*.*

Filterisasi data pada dataset kebakaran sangat penting untuk menjamin kualitas dari model objek lokalisasi yang akan dibangun. Proses ini meliputi pemilihan data yang sesuai dengan kriteria yang ditentukan dan penghapusan data yang tidak sesuai atau tidak valid.kebakaran serta label yang sesuai dengan objek tersebut, dalam hal ini menghAnnotasi yang baik akan membantu model untuk belajar dan mengenali objek kebakaran dengan baik.

Beberapa kriteria yang dapat digunakan dalam filterisasi data adalah ukuran bounding box, label objek, dan kualitas gambar. Ukuran bounding box yang terlalu kecil atau terlalu besar dapat dihapus karena tidak representatif. Label objek yang tidak sesuai dengan objek yang akan dideteksi juga harus dihapus. Kualitas gambar yang buruk atau tidak jelas juga harus dihapus untuk menjamin kualitas data yang digunakan.

Selain itu, data yang digunakan harus juga diperiksa secara manual untuk memastikan bahwa semua data yang digunakan sesuai dengan kriteria yang ditentukan dan tidak ada data yang tidak valid. Dengan filterisasi data yang baik, model objek lokalisasi yang akan dibangun akan lebih baik dan akurat dalam mengenali objek kebakaran.

Pengambilan data dilakukan dengan menggabungkan dataset api dan kebakaran, dengan total dataset mencapai 996 gambar setelah dilakukan filterisasi. Data tersebut terdiri atas 403 gambar api berukuran kecil dan 262 gambar api berukuran besar, pada dataset asap terdiri atas 203 gambar asap berwarna hitam dan 128 gambar asap berwarna putih.

##### Pembagian Dataset

## Training

Proses training di bagi ke dalam dua model Deep Learning yakni YOLOv4 dan YOLOv4-tiny, dengan membagi proses training ke dalam tiga dataset berbeda, yakni dataset yang berisikan asap, dataset yang berisikan api dan dataset yang berisikan keduanya, pembagian training ini bertujuan untuk mengetahui model mana yang memiliki akurasi, precission dan recall terbesar serta kecepatan inferensi terbaik dari ketiga model.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Parameter | Fire | | Smoke | | Fire + Smoke | |
| YOLOv4 | YOLOv4  tiny | YOLOv4 | YOLOv4  tiny | YOLOv4 | YOLOv4  tiny |
| Accuracy |  |  |  |  |  |  |
| Precission |  |  |  |  |  |  |
| Recall |  |  |  |  |  |  |
| F1-Score |  |  |  |  |  |  |
| FPS |  |  |  |  |  |  |
| Size |  |  |  |  |  |  |

## Implementasi

# Referensi

[1] U. Nations, “World Population Prospects 2019: Data Booklet,” Jun. 2019, doi: 10.18356/3E9D869F-EN.

[2] delli noviarti and C. chairil, “ANALISIS SISTEM PROTEKSI KEBAKARAN  DI KAWASAN PADAT PENDUDUK  (STUDI KASUS PADA KELURAHAN KERTAPATI PALEMBANG),” *TEKNIKA: Jurnal Teknik*, vol. 5, no. 2, pp. 117–129, Jan. 2019, doi: 10.35449/TEKNIKA.V5I2.97.

[3] D. Winberg, “International Fire Death Rate Trends,” 2016.

[4] G. Kurlick, “Stop, drop, and roll: Workplace hazards of local government firefighters, 2009,” *Monthly labor review / U.S. Department of Labor, Bureau of Labor Statistics*, vol. 135, pp. 18–25, Aug. 2012.

[5] S. Ramasubramanian, S. A. Muthukumaraswamy, and A. Sasikala, “Fire Detection using Artificial Intelligence for Fire-Fighting Robots,” *Proceedings of the International Conference on Intelligent Computing and Control Systems, ICICCS 2020*, pp. 180–185, May 2020, doi: 10.1109/ICICCS48265.2020.9121017.

[6] J. Guo, Z. Hou, X. Xie, S. Yao, Q. Wang, and X. Jin, “Faster R-CNN Based Indoor Flame Detection for Firefighting Robot,” *2019 IEEE Symposium Series on Computational Intelligence, SSCI 2019*, pp. 1390–1395, Dec. 2019, doi: 10.1109/SSCI44817.2019.9002843.

[7] H. D. Duong and D. T. Tinh, “An efficient method for vision-based fire detection using SVM classification,” *2013 International Conference on Soft Computing and Pattern Recognition, SoCPaR 2013*, pp. 190–195, 2013, doi: 10.1109/SOCPAR.2013.7054125.

[8] A. Q. Nguyen, H. T. Nguyen, V. C. Tran, H. X. Pham, and J. Pestana, “A Visual Real-time Fire Detection using Single Shot MultiBox Detector for UAV-based Fire Surveillance,” *ICCE 2020 - 2020 IEEE 8th International Conference on Communications and Electronics*, pp. 338–343, Jan. 2021, doi: 10.1109/ICCE48956.2021.9352080.

[9] R. Xu, H. Lin, K. Lu, L. Cao, and Y. Liu, “A Forest Fire Detection System Based on Ensemble Learning,” *Forests 2021, Vol. 12, Page 217*, vol. 12, no. 2, p. 217, Feb. 2021, doi: 10.3390/F12020217.

[10] M. Kanwar and L. Agilandeeswari, “IOT Based Fire Fighting Robot,” *2018 7th International Conference on Reliability, Infocom Technologies and Optimization: Trends and Future Directions, ICRITO 2018*, pp. 718–723, Aug. 2018, doi: 10.1109/ICRITO.2018.8748619.

[11] S. Zhao, B. Liu, Z. Chi, T. Li, and S. Li, “Characteristics Based Fire Detection System Under the Effect of Electric Fields With Improved Yolo-v4 and ViBe,” *IEEE Access*, vol. 10, pp. 81899–81909, 2022, doi: 10.1109/ACCESS.2022.3190867.

[12] M. Kulshreshtha, S. S. Chandra, P. Randhawa, G. Tsaramirsis, A. Khadidos, and A. O. Khadidos, “OATCR: Outdoor Autonomous Trash-Collecting Robot Design Using YOLOv4-Tiny,” *Electronics 2021, Vol. 10, Page 2292*, vol. 10, no. 18, p. 2292, Sep. 2021, doi: 10.3390/ELECTRONICS10182292.

[13] M. Madhar, “RANCANG BANGUN SISTEM MONITORING DETEKSI DINI KEBAKARAN DENGAN FITUR GPS BERBASIS WEBSITE,” *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 2, no. 1, pp. 367–372, Mar. 2018, doi: 10.36040/JATI.V2I1.1692.

[14] S. Suwarjono *et al.*, “Design of a Home Fire Detection System Using Arduino and SMS Gateway,” *Knowledge 2021, Vol. 1, Pages 61-74*, vol. 1, no. 1, pp. 61–74, Nov. 2021, doi: 10.3390/KNOWLEDGE1010007.

[15] A. Anton, N. F. Nissa, A. Janiati, N. Cahya, and P. Astuti, “Application of Deep Learning Using Convolutional Neural Network (CNN) Method for Womenâ€TMs Skin Classification,” *Scientific Journal of Informatics*, vol. 8, no. 1, pp. 144–153, May 2021, doi: 10.15294/SJI.V8I1.26888.

[16] S. Mostafa and F. X. Wu, “Diagnosis of autism spectrum disorder with convolutional autoencoder and structural MRI images,” *Neural Engineering Techniques for Autism Spectrum Disorder: Volume 1: Imaging and Signal Analysis*, pp. 23–38, Jan. 2021, doi: 10.1016/B978-0-12-822822-7.00003-X.

[17] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, “Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 658–666, Feb. 2019, doi: 10.48550/arxiv.1902.09630.

[18] O. J. M. Al-Furaiji, N. A. Tuan, and T. V. Yurevich, “A new fast efficient non-maximum suppression algorithm  based on image segmentation,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 19, no. 2, pp. 1062–1070, Aug. 2020, doi: 10.11591/IJEECS.V19.I2.PP1062-1070.

[19] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-December, pp. 779–788, Dec. 2016, doi: 10.1109/CVPR.2016.91.

[20] A. Mikołajczyk and M. Grochowski, “Data augmentation for improving deep learning in image classification problem,” *2018 International Interdisciplinary PhD Workshop, IIPhDW 2018*, pp. 117–122, Jun. 2018, doi: 10.1109/IIPHDW.2018.8388338.

[21] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection,” Apr. 2020, doi: 10.48550/arxiv.2004.10934.

[22] D. M. W. Powers and Ailab, “Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation,” Oct. 2020, doi: 10.48550/arxiv.2010.16061.

[23] M. Everingham, L. van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The Pascal Visual Object Classes (VOC) Challenge,” *International Journal of Computer Vision 2009 88:2*, vol. 88, no. 2, pp. 303–338, Sep. 2009, doi: 10.1007/S11263-009-0275-4.

[24] E. Zhang and Y. Zhang, “Eleven Point Precision-recall Curve,” *Encyclopedia of Database Systems*, pp. 981–982, 2009, doi: 10.1007/978-0-387-39940-9\_481.