

BAB 2

LANDASAN TEORI

2.1 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:

Tabel 2.1 State of Art Penelitian

No .	Judul Jurnal	Tipe	Ringkasan
1	M. Kanwar and L. Agilandeewari, "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	Deteksi Api dengan YoloV4	Mendeteksi api yang muncul akibat penerapan perlakuan efek medan elektrik yang berbeda menggunakan YoloV4.

	ViBe," in IEEE Access, vol. 10, pp. 81899-81909, 2022, doi: 10.1109/ACCESS.2022.3190867. [11]		
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 <i>driver</i> motor L298N serta <i>stepper</i> 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,	Sistem Alerting	Sistem <i>alerting</i> yang terhubung dengan sensor api dan Arduino UNO. Intensitas pembacaan pada sensor api apabila melebihi

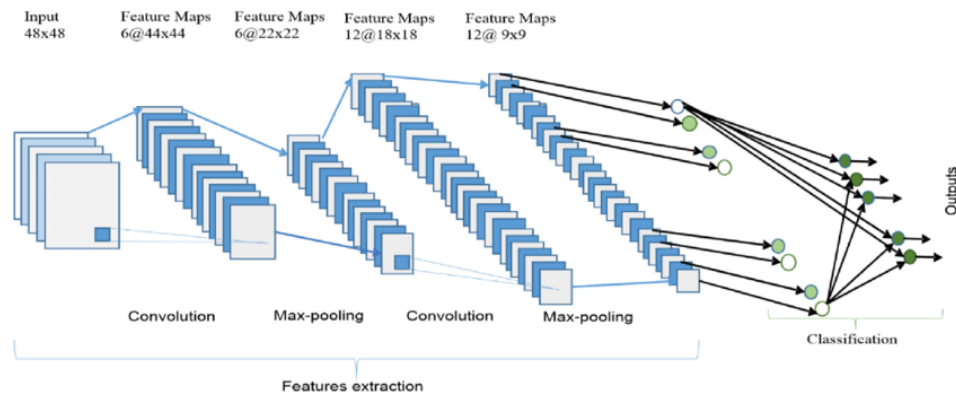
	Wahyu Caesarendra, Stanislaw Legutko, and Adam Glowacz. 2021. "Design of a Home Fire Detection System Using Arduino and SMS Gateway" Knowledge 1, no. 1: 61-74. https://doi.org/10.3390/knowledge1010007 . [14]		batasan tertentu maka <i>buzzer</i> akan berbunyi dan mengirimkan pesan peringatan melalui omodul GSM.
--	---	--	--

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

2.2 *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,

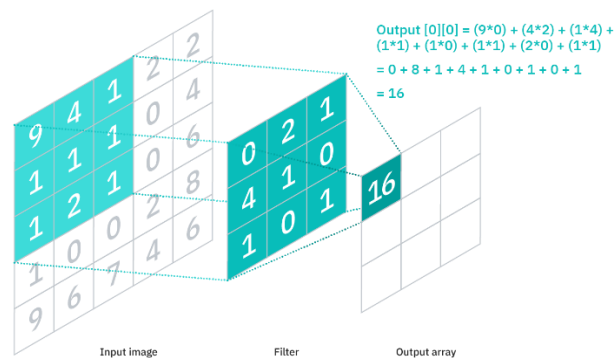
operasi linier menggunakan operasi konvolusi, sedangkan beratnya tidak lagi satu dimensi tetapi empat dimensi yang merupakan kumpulan dari konvolusi kernel. Sebuah CNN terdiri dari beberapa jenis layer, yaitu *Convolutional Layer*, *Max Pooling Layer*, dan *Fully Connected Layer* [15].



Gambar 2.1 Arsitektur Convolutional Neural Network

1. *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].

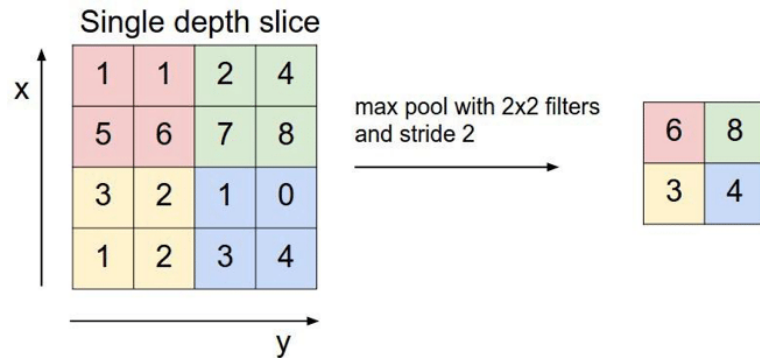


Gambar 2.2 Lapisan Konvolusi

2. *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

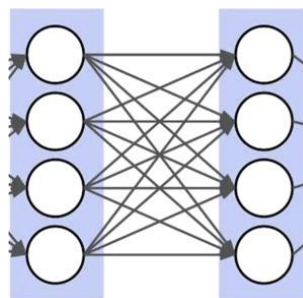
pooling atau mengambil nilai terbesar dari bagian tersebut. Namun terdapat metode pooling lain yang dapat digunakan seperti average pooling atau L2-norm pooling.



Gambar 2.3 Lapisan *Max Pooling*

3. *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.



Gambar 2.4 Lapisan *Fully Connected*

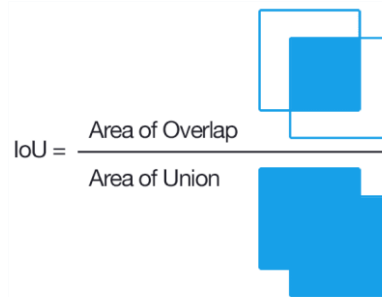
2.3 YOLOv4

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*

1. *Intersection over Union (IoU)*

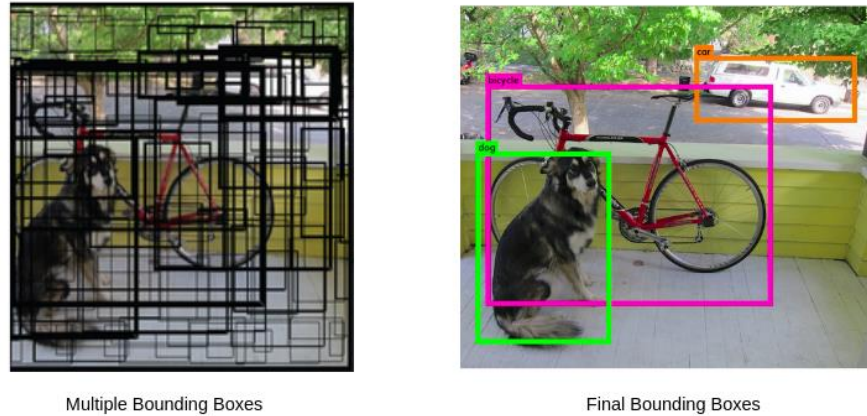
IoU, juga dikenal sebagai indeks Jaccard, adalah yang metrik yang paling umum untuk membandingkan kesamaan antara dua bentuk arbitrer. IoU mengkodekan properti bentuk dari objek yang dibandingkan, yakni lebar, tinggi, dan lokasi dua *bounding boxes*, ke dalam properti wilayah dan kemudian menghitung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



Gambar 2.5 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*.

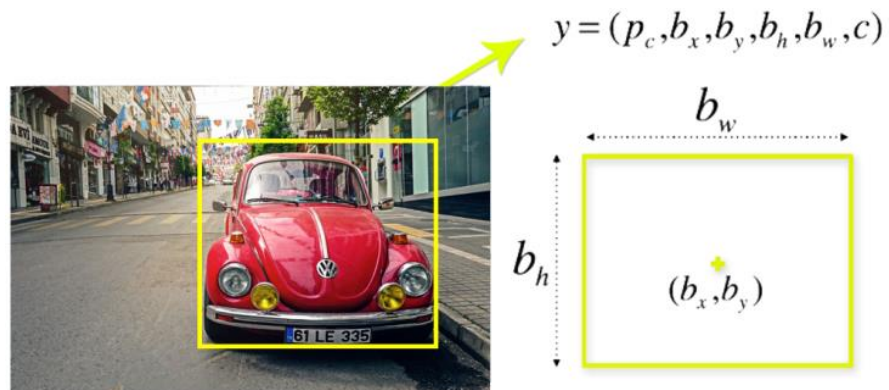
Pada proses deteksi objek dalam gambar, *bounding boxes* seringkali terbentuk. Algoritma *Non-max suppression (NMS)* digunakan untuk mengeliminasi *bounding boxes* terbaik dari gambar. NMS merupakan algoritma yang bertugas untuk mencari local maxima berdasarkan nilai batasan IoU awal model [18] Apabila 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 dilanjutkan dengan proses eliminasi kedua dengan mengambil *bounding boxes* dengan *confidence score* terbesar dari *bounding boxes*.



Gambar 2.6 *Non-max Suppression (NMS)*

2. *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.



Gambar 2.7 *Object Localization dan Classification*

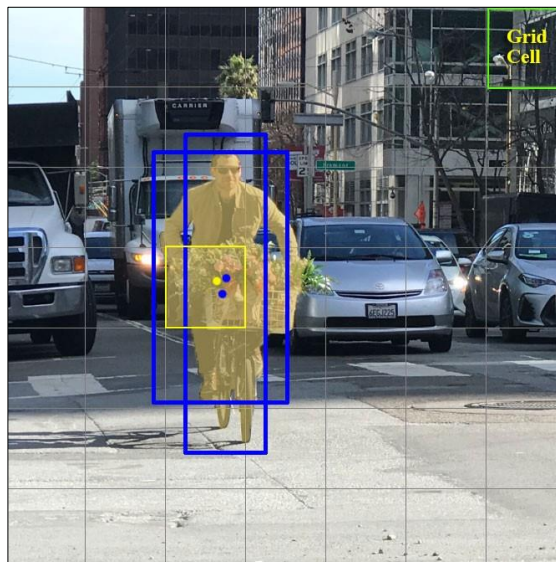
Output dari lokalisasi model akan mencakup informasi dari bounding boxes, sehingga *output* akan terlihat pada vektor dibawah ini:

ρ_c : *Confidence score* dari objek pada *bounding boxes*

$$y = \begin{bmatrix} \rho_c \\ t_x \\ t_y \\ t_h \\ t_w \\ c_1 \\ \dots \\ c_n \end{bmatrix} \quad \begin{array}{ll} b_x: & \text{Pusat koordinat } x \text{ terhadap bagian kiri} \\ & \text{bounding boxes} \\ b_y: & \text{Pusat koordinat } y \text{ terhadap bagian atas} \\ & \text{bounding boxes} \\ b_h: & \text{Tinggi bounding boxes} \\ b_w: & \text{Lebar bounding boxes} \\ c_n: & \text{Bernilai satu apabila terdapat kelas-n yang} \\ & \text{terdeteksi, nol apabila tidak terdapat kelas-n} \\ & \text{yang terdeteksi} \end{array}$$

3. Grid Cells

YOLO membagi masukan gambar ke dalam $S \times S$ grid. Apabila pusat dari objek terletak pada *grid cell* maka *grid cell* tersebut 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*.



Gambar 2.8 *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:

$$\text{Box Confidence Score} = P_r(\text{Object}) \times \text{IoU}$$

$P_r(\text{Object})$: Probabilitas dari *bounding boxes* terdapat objek

IoU : *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.

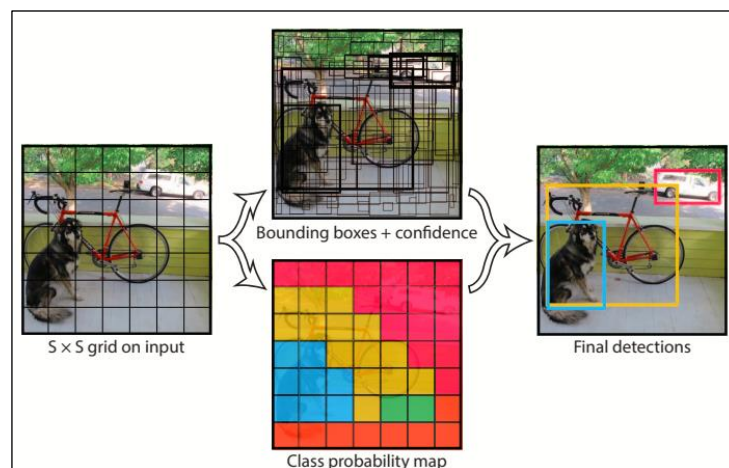
$$\text{Class Probability} = P_r(\text{Class}_i | \text{Object})$$

$$\text{Class Confidence Score} = P_r(\text{Class}_i) \times \text{IoU}$$

$P_r(\text{Class}_i | \text{Object})$: Probabilitas suatu objek berada di Class_i ketika suatu objek terdeteksi

IoU : *Intersection of Union* (IoU)

$P_r(\text{Class}_i)$: Probabilitas objek itu berada di Class_i



Gambar 2.9 Tahapan deteksi menggunakan Algoritma YOLO

4. *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*.

5. *Recall*, *Precision* dan *mAP*

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

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

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

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

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

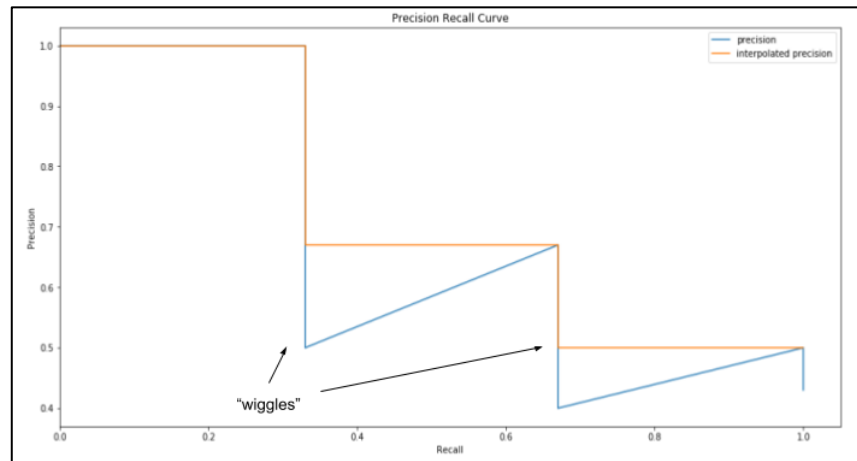
$$AP = \int_0^1 p(r) dr$$

$p(r)$: Fungsi *precision*
 terhadap *recall*
 r : Nilai *recall*

Proses interpolasi dilakukan terhadap nilai PR untuk meminimalisir variasi yang ditimbulkan akibat urutan sampel. Interpolasi AP dilakukan dengan mengganti nilai *recall* \hat{r} dengan maksimum *precision* pada *recall* $recall \geq \tilde{r}$ [23]

$$p_{interpolasi} = \max_{\tilde{r} \geq r} p(\tilde{r})$$

$p_{interpolasi}$: *Precision* yang telah terinterpolasi
 \tilde{r} : Nilai *recall* dengan *precision* terbesar



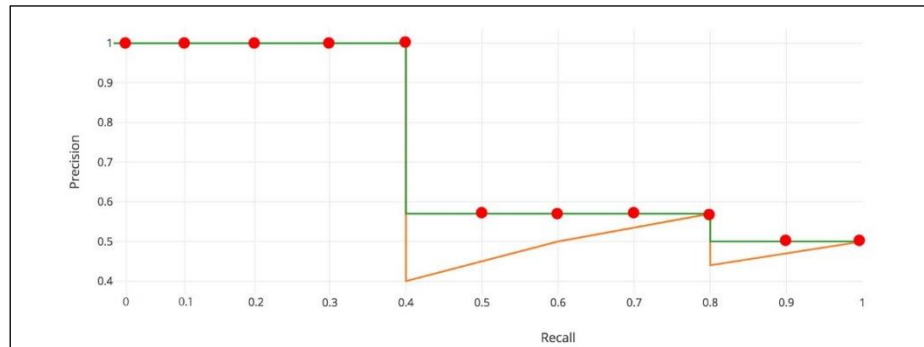
Gambar 2.10 Nilai interpolasi AP berdasarkan nilai maksimum antara *Recall-Precision*

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 *precision* dapat dirumuskan ke dalam persamaan berikut [24] :

$$mAP = \frac{1}{N} \sum_1^N p_{interpolasi}(r)$$

N : Nilai *recall* pada interval [0, 1]
mAP : *mean Average-Precision*
 $p_{interpolasi}(r)$: Nilai *precision* yang telah ter-interpolasi



Gambar 2.11 *mean Average-Precision* (mAP)

2.4 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, Raspberry Pi Foundation 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 dilengkapi 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.