

Objek Deteksi Makanan Khas Palembang Menggunakan Algoritma *YOLO (You Only Look Once)*

Lusiana Rahma, Hadi Syaputra, A.Haidar Mirza & Susan Dian Purnamasari

Fakultas Teknik Ilmu Komputer, Universitas Bina Darma

Email: lusianarahma9@gmail.com

Abstract

Deep learning is a part of machine learning method that uses artificial neural network (ANN). The type of learning in deep learning can be supervised, semi-supervised, and unsupervised [7]. CNN & RNN (Supervised) and RBM & Autoencoder (Unsupervised) are deep learning algorithms. All of the above algorithms have uses in their respective fields, depending on what we want to use them for. One of the most frequently used cases for deep learning is object detection and classification. The Convolutional Neural Network (CNN) algorithm is the most widely used algorithm for object detection cases, one of the reasons because it is supported by Google's Tensorflow framework, but it turns out that there is one object detection algorithm that has a higher level of accuracy and processing speed, namely You Only Look Once (YOLO) which can run on 2 frameworks (Darknet & Darkflow) and is supported by GPU. That's why here the author prefers to do object detection with the You Only Look Once (YOLO) method. The research data with the title Palembang Food Detection Object Using the YOLO (You Only Look Once) Algorithm is a sample photo of food from Google Image. There are 31 types of Palembang specialties, each type consists of approximately 50 to 70 images, so the total images used are from 31 types of Palembang foods, namely 1955 images with jpeg format for training data, and 31 images with jpeg format typical Palembang foods for test data.

Keywords : Traditional food of Palembang, Deep Learning, YOLO, image processing

1. Pendahuluan

Salah satu bagian dari metode *machine learning* adalah *deep learning* yang menggunakan *artificial neural network* (ANN). Jenis pembelajaran dalam *deep learning* dapat berupa *supervised*, *semi-supervised*, dan *unsupervised* [7].

CNN & RNN (Supervised) dan RBM & Autoencoder (Unsupervised) merupakan algoritma dari *deep learning*. Semua algoritma diatas mempunyai kegunaan di bidang masing-masing, tergantung kita ingin menggunakan itu semua untuk kasus apa. Salah satu kasus paling sering digunakan untuk deep learning adalah pendeteksian dan pengklasifikasian objek (Object Detection). Algoritma Convolutional Neural Network (CNN) merupakan algoritma yang paling banyak dipakai untuk kasus object detection, salah satu alasannya karena didukung oleh framework Tensorflow buatan Google, tetapi ternyata ada satu algoritma object detection yang mempunyai tingkat akurasi dan kecepatan proses yang lebih tinggi, yaitu You Only Look Once (YOLO) yang dapat dijalankan di 2 framework (Darknet & Darkflow) dan didukung oleh GPU (Alfarisi, 2021) (Alfarisi, 2021). Karena itulah disini Penulis lebih memilih melakukan object detection dengan metode You Only look Once (YOLO).

Metode You Only Look Once (YOLO) merupakan salah satu metode yang paling cepat dan akurat pada pendeteksian objek bahkan mampu melebihi hingga 2 kali kemampuan algoritma lain. YOLO mempunyai banyak versi yang sering diterapkan yaitu mulai versi YOLO, YOLOv2 hingga yang terbaru saat ini adalah YOLOv3. Dalam penelitian ini,

dengan meningkatkan YOLOv2 model yang disebut YOLOv3_Vehicle diusulkan untuk mendeteksi kendaraan. Untuk mendapatkan kotak jangkar yang lebih baik, kotak pembatas kendaraan pada dataset pelatihan dikelompokkan dengan cluster sarana ++, dan memilih 6 kotak jangkar dengan ukuran yang berbeda .

Palembang sebagai kota di Indonesia, menjadi tempat berkumpulnya penduduk dari berbagai budaya yang memiliki beranekaragam komunitas seperti Tionghoa, India dan Arab Karena beranekaragaman itula, maka secara langsung maupun tidak langsung, membawa keanekaragaman kuliner tiap-tiap budaya juga dipalembang. Contohnya makanan khas seperti pempek, tekwan, model yang dikenal masyarakat pada umumnya, serta dikarenakan kurangnya pemahaman mengenai berbagai kuliner khas palembang maka diperlukan sebuah algoritma yang dikembangkan untuk mendeteksi sebuah objek secara *real-time*. Sistem pendeteksian yang dilakukan adalah dengan menggunakan *repurpose classifier* atau *localizer* untuk melakukan deteksi.

Berdasarkan uraian yang telah dijelaskan diatas maka penulis tertarik meneliti “Objek Deteksi Makanan Khas Palembang Menggunakan Algoritma YOLO(You Only Look Once)” yang terdiri dari 31 jenis makanan yang terbagi kedalam 3 kategori diantaranya : makanan ringan, makanan sedang, dan makanan berat khas palembang.

2. Tinjauan Literatur

2.1 Citra Digital

Citra adalah suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek. Citra terbagi 2 yaitu citra yang bersifat analog dan ada citra yang bersifat digital. Citra analog adalah citra yang bersifat *continue* seperti gambar pada monitor televisi, foto sinar X, dan lain-lain. Sedangkan pada citra digital adalah citra yang dapat diolah oleh komputer (Antoni, Fikari & Akbar, 2018; Antoni & Akbar, 2019; Antoni, Jie & Abareshi, 2020; Antoni, Herdiansyah, Akbar & Sumitro, 2021).

Citra digital adalah representatif dari citra yang diambil oleh mesin dengan bentuk pendekatan berdasarkan sampling dan kuantisasi. Sampling menyatakan besarnya kotak-kotak yang disusun dalam baris dan kolom. Dengan kata lain, sampling pada citra menyatakan besar kecilnya ukuran *pixel* (titik) pada citra, dan kuantisasi menyatakan besarnya nilai tingkat kecerahan yang dinyatakan dalam nilai tingkat keabuan (*grayscale*) sesuai dengan jumlah bit biner yang digunakan oleh mesin, dengan kata lain kuantisasi pada citra menyatakan jumlah warna yang ada pada citra (*Pengolahan Citra Digital - KajianPustaka.Com*, n.d.).

2.2 Deep Learning

Deep learning adalah seperangkat algoritma dalam *machine learning* yang berusaha belajar dalam berbagai level, sesuai dengan tingkat abstraksi yang berbeda. Ini biasanya menggunakan jaringan syaraf tiruan. Tingkat dalam model statistik yang dipelajari ini sesuai dengan tingkat konsep yang berbeda, di mana konsep tingkat yang lebih tinggi ditentukan dari tingkat yang lebih rendah, dan konsep tingkat yang lebih rendah dapat membantu untuk mendefinisikan banyak konsep tingkat yang lebih tinggi (“Apakah Deep Learning?,” n.d.).

2.3 Precision Dan Recall

Recall dapat diartikan sebagai metode pengujian yang membandingkan jumlah informasi relevan yang didapatkan sistem dengan jumlah seluruh informasi relevan yang ada dalam koleksi informasi (baik yang terambil atau tidak terambil oleh sistem).

Sedangkan *Precision* dapat diartikan sebagai kepersisan atau kecocokan (antara permintaan informasi dengan jawaban terhadap permintaan itu). Jika seseorang mencari informasi di sebuah sistem, dan sistem menawarkan beberapa dokumen, maka kepersisan ini sebenarnya juga adalah relevansi. Artinya, seberapa persis atau cocok dokumen tersebut untuk keperluan pencari informasi, bergantung pada seberapa relevan dokumen tersebut bagi si pencari (Lestari, n.d.).

Perolehan (*recall*) berhubungan dengan kemampuan sistem untuk memanggil dokumen yang relevan. Untuk menghitung nilai perolehan (*recall*) digunakan rumus sebagai berikut:

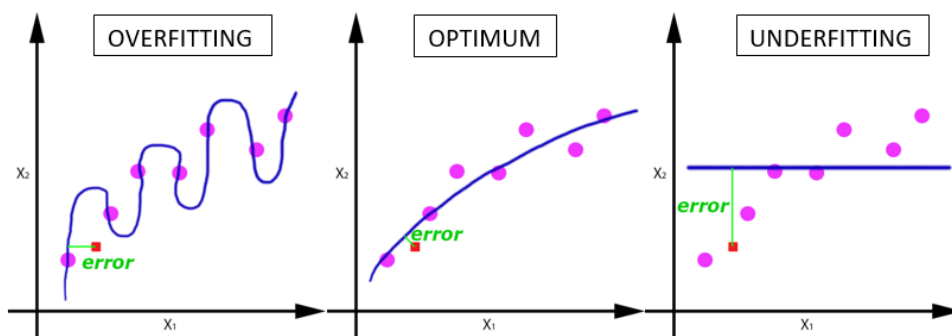
$$\text{Recall (R)} = \frac{\text{Jumlah Dokumen Relevan yang Terambil}}{\text{Jumlah Dokumen Relevan dalam Database}}$$

Ketepatan (*Precision*) berkaitan dengan kemampuan sistem untuk tidak memanggil dokumen yang tidak relevan. Untuk menghitung nilai ketepatan (*precision*) digunakan rumus sebagai berikut :

$$\text{Precision (P)} = \frac{\text{Jumlah Dokumen Relevan yang Terambil}}{\text{Jumlah Dokumen Relevan dalam Pencarian}}$$

2.4 Epoch

Epoch adalah ketika seluruh dataset sudah melalui proses training pada Neural Network sampai dikembalikan ke awal untuk sekali putaran, karena satu *Epoch* terlalu besar untuk dimasukkan (*feeding*) kedalam komputer maka dari itu kita perlu membaginya kedalam satuan kecil (*batches*). Satu epoch mengarah pada *underfitting* pada grafik (di bawah).



Gambar 2.1 Epoch

Seiring bertambahnya jumlah epoch, semakin banyak pula weight (bobot) yang berubah dalam Neural Network dan kurvanya melengkung dari kurva yang kurang sesuai hingga selaras dengan kurva yang overfitting (*Memahami Epoch Batch Size Dan Iteration - JournalToday*, n.d.).

2.5 Python

Python adalah bahasa pemrograman model skrip (scripting language) yang berorientasi obyek. Python dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai platform sistem operasi. Python merupakan bahasa pemrograman yang freeware atau perangkat bebas dalam arti sebenarnya, tidak ada batasan dalam penyalinannya atau mendistribusikannya. Lengkap dengan source codenya, debugger dan profiler, antarmuka yang terkandung di dalamnya untuk pelayanan antarmuka, fungsi sistem, GUI (antarmuka pengguna grafis), dan basis datanya (*Perkasa et al.*, n.d.).

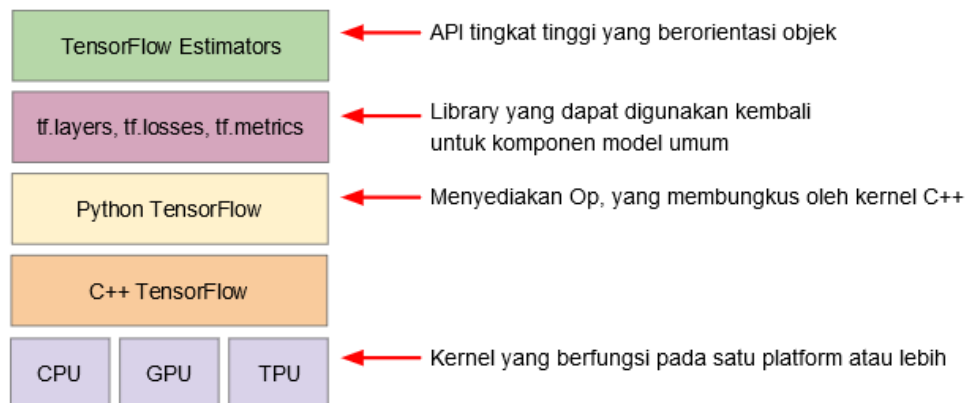
2.6 Keras

Dilansir dari halaman rpubs.com (*RPubs - Klasifikasi Gambar Dengan Framework Keras*, n.d.) dijelaskan bahwa keras merupakan *high-level neural network API* yang dikembangkan dengan *Python* dengan fokus tujuan untuk mempercepat proses riset atau percobaan. Beberapa fitur utama dari Keras :

1. Mampu menjalankan *source code* yang sama menggunakan CPU atau GPU dengan lancar
2. API yang *user-friendly* sehingga mempermudah penggunaanya dalam proses prototipe model deep learning
3. Dukungan *built-in* untuk CNN atau Convolutional Neural Networks (Computer Vision), RNN atau Recurrent Neural Networks (untuk *sequence processing*), dan kombinasi keduanya
4. Dapat digunakan untuk hampir semua jenis dari model deep learning

2.7 Tensorflow

TensorFlow adalah pustaka perangkat lunak sumber terbuka dan gratis untuk pembelajaran mesin. TensorFlow dapat digunakan dalam berbagai tugas tetapi memiliki fokus khusus pada pelatihan dan inferensi jaringan neural dalam. TensorFlow adalah pustaka matematika simbolis berdasarkan dataflow dan pemrograman. Saat ini, TensorFlow merupakan pustaka pembelajaran meesin paling terkenal di dunia (*Lina*, 2021).



Gambar 2.2 Hirarki Toolkit *TensorFlow*

(Sumber : (Panduan Awal TF, n.d.))

Berikut adalah ringkasan tujuan dari berbagai lapisan :

Tabel 2.1 Tabel Tujuan

Toolkit	Desjripsi
Estimator (tf.estimator)	API tingkat tinggi yang berorientasi objek.
tf.layers/tf.losses/tf.metrics	Library untuk komponen model umum.
TensorFlow	API dengan tingkat lebih rendah

TensorFlow terdiri dari dua komponen berikut:

1. uffer protokol grafik
2. waktu proses yang menjalankan grafik (terdistribusi)

2.8 Open CV

OpenCV (Open Source Computer Vision Library), adalah sebuah library open source yang dikembangkan oleh intel yang fokus untuk menyederhanakan proqraming terkait citra digital. Di dalam OpenCV sudah mempunyai banyak fitur, antara lain : pengenalan wajah, pelacakan wajah, deteksi wajah, Kalman filtering, dan berbagai jenis metode AI (Artificial Intelligence). Dan menyediakan berbagai algoritma sederhana terkait Computer Vision untuk low level API (*Introduction to Open CV / BINUS UNIVERSITY MALANG / Pilihan Universitas Terbaik Di Malang, n.d.*).

2.9 Google Colaboratory

Google Colaboratory atau disebut juga *Colab* adalah tools baru yang dikeluarkan oleh yang dibuat oleh *Google Internal Research*. Tools ini membantu para *Researcher* dalam mengolah data untuk keperluan belajar pada pengolahan data menggunakan *Machine Learning*. *Penggunaan tools ini mirip dengan Jupyter Notebook dimana tools ini dibuat*

diatas environment Jupyter . Google Colab memiliki penyimpanan berupa Google Drive dimana tools ini berjalan dengan sistem Cloud. Google Colab ini menyediakan layanan GPU gratis kepada penggunanya sebagai backend komputasi dan dapat digunakan selama 12 jam pada suatu waktu (Digmi, 2018).

2.10 Confusion Matrick

Confusion Matrix adalah pengukuran performa untuk masalah klasifikasi *machine learning* dimana keluaran dapat berupa dua kelas atau lebih. *machine learning* merupakan proses komputer untuk belajar dari data. *Confusion Matrix* juga merupakan *tools* analitik prediktif yang menampilkan dan membandingkan nilai aktual atau nilai sebenarnya dengan nilai hasil prediksi model yang dapat digunakan untuk menghasilkan metrik evaluasi seperti *Accuracy* (akurasi), *Precision*, *Recall*, dan *F1-Score* atau *F-Measure* (*Confusion Matrix*, n.d.). Tools ini menghasilkan 4 nilai yang berbeda yaitu :

True Positive (TP) : Jumlah data yang bernilai Positif dan diprediksi benar sebagai Positif.

False Positive (FP) : Jumlah data yang bernilai Negatif tetapi diprediksi sebagai Positif.

False Negative (FN) : Jumlah data yang bernilai Positif tetapi diprediksi sebagai Negatif.

True Negative (TN) : Jumlah data yang bernilai Negatif dan diprediksi benar sebagai Negatif.

		Actual Values	
		1 (Postive)	0 (Negative)
Predicted Values	1 (Postive)	TP (True Positive)	FP (False Positive) Type I Error
	0 (Negative)	FN (False Negative) Type II Error	TN (True Negative)

Gambar 2.3 *Confusion Matrick*

2.11 Convolutional Neural Network (CNN)

CNN merupakan operasi konvolusi yang menggabungkan beberapa lapisan pemrosesan, menggunakan beberapa elemen yang beroperasi secara paralel dan terinspirasi oleh sistem saraf biologis. Pada CNN setiap neuron dipresentasikan dalam bentuk 2 dimensi, sehingga metode ini cocok untuk pemrosesan dengan input berupa citra (Arrofiqoh & Harintaka, 2018).

CNN bekerja secara hierarki, sehingga output pada lapisan konvolusi pertama digunakan sebagai input pada lapisan konvolusi selanjutnya. Pada proses klasifikasi terdiri dari fully-connected dan fungsi aktivasi (softmax) yang outputnya berupa hasil klasifikasi (Katole et al., 2015).

2.12 Algoritma Yolo

You Only Look Once (Yolo) adalah sebuah algoritma yang dikembangkan untuk mendeteksi sebuah objek secara real-time. Sistem pendeteksian yang dilakukan adalah dengan menggunakan *repurpose classifier* atau *localizer* untuk melakukan deteksi. sebuah model diterapkan pada sebuah citra di beberapa lokasi dan skala. Daerah dengan citra yang diberi *score* paling tinggi akan dianggap sebagai sebuah pendeteksian (Unsky, 2017).

Yolo menggunakan pendekatan jaringan saraf tiruan (JST) untuk mendeteksi objek pada sebuah citra. Jaringan ini membagi citra menjadi beberapa wilayah dan memprediksi setiap kotak pembatas dan probabilitas untuk setiap wilayah. Kotak-kotak pembatas ini kemudian dibandingkan dengan probabilitas yang diprediksi. *Yolo* memiliki beberapa kelebihan dibandingkan dengan sistem yang berorientasi pada *classifier*, terlihat dari seluruh citra pada saat dilakukan *test* dengan prediksi yang diinformasikan secara global pada citra (Redmon, 2016). Hal tersebut juga sintesis jaringan saraf ini tidak seperti sistem Region-Convolutional Neural Network (R-CNN) yang membutuhkan ribuan untuk sebuah citra sehingga membuat *Yolo* lebih cepat hingga beberapa kali daripada R-CNN.

Beberapa penelitian yang mempunyai hubungan dengan penelitian ini adalah Penelitian berjudul Pengenalan Objek Makanan Cepat Saji Pada Video Dan *Real Time webcam* Menggunakan Metode *You Look Only Once* (YOLO) yang dilakukan oleh (Karlina & Indarti, 2019). Tujuan dari penelitian tersebut adalah untuk pengenalan objek pada citra makanan cepat saji menggunakan YOLO. Penelitian tersebut terdiri dari beberapa tahap yaitu pengumpulan data, pra-proses data, konfigurasi jaringan YOLO, pelatihan model YOLO dan pengujian. Jumlah data citra yang digunakan dalam pelatihan yaitu 468 citra yang terdiri dari tiga jenis makanan cepat saji. Nilai *avgloss* pada model akhir yang dibangun dengan YOLO yaitu 4.6% dan nilai validasi *mAP@100%*. Berdasarkan hasil pengujian yang dilakukan menggunakan video dan real time webcam, objek pada citra makanan cepat saji berhasil dikenali dengan akurasi 63% sampai 100%.

Penelitian berjudul Pembuatan Pendeteksi Obyek Dengan Metode *You Only Look Once* (YOLO) Untuk *Automated Teller Machine* (ATM) yang dilakukan oleh (Wasril et al., n.d.). Didalam penelitian tersebut dikatakan bahwa tujuan dari pembuatan pendeteksi objek adalah untuk mendeteksi benda-benda yang dilarang digunakan ketika berada di ATM. Salah satu metode yang digunakan untuk menciptakan *Object Detection* yaitu *You Only Look Once* (YOLO). Implementasi ide ini tersedia pada *DARKNET (open source neural network)*.

3. Metode Penelitian

3.1 Metode Pengumpulan Data

3.1.1 Populasi Dan Sampel (Subjek Penelitian)

Populasi dalam penelitian ini adalah makanan khas Palembang yang semua datanya adalah makanan khas Palembang. Sedangkan untuk sampel dalam penelitian ini adalah makanan ringan sebanyak 16, makanan sedang 11 sebanyak , dan makanan berat sebanyak 4.

3.1.2 Teknik Sampling

Jenis data yang digunakan dalam penelitian ini adalah data sekunder. “Data sekunder adalah jenis data tambahan yang tidak diperoleh dari sumber utama, tetapi sudah melalui sumber kesekian (Fauzi, Dencik & Asiati, 2019). Artinya, orang-orang tersebut tidak merasakan secara langsung fenomena yang sedang diteliti, tetapi mendapatkan informasinya dari sumber-sumber primer lainnya” (Hakim, 2020). Dilansir dari halaman scribd.com bahwa syarat terpenting yang perlu diperhatikan dalam pengambilan sampel ada dua macam yaitu :

- 1) Jumlah sampel yang mencukupi
- 2) Profil sampel yang dipilih harus mewakili

Berdasarkan penjelasan diatas, maka dapat disimpulkan bahwa penentuan sampel yang diambil adalah sebagai berikut :

- 1) Makanan ringan khas Palembang yaitu pempek lenjer, pempek keriting, pempek adaan, pempek kulit, pempek tahu, pempek pistol, pempek panggang, pempek belah, otak-otak, kue maksubah, kue delapan jam, kue srikaya, kue ragit, kue gandum, enggak ketan, dan lapis kojo.
- 2) Makanan sedang khas Palembang yaitu lenggang, tekwan, model ikan, model gandum, laksanakan, celimpungan, mie celor, burgo, lakso, pempek kapal selam, dan rujak mie.
- 3) Makanan berat khas Palembang yaitu pindang paten, pindang tulang, pindang paten tempoyak, dan malbi.

Dengan menggunakan jumlah sampel yang sama memungkinkan untuk lebih mempermudah dalam mempartisi data dalam mengklasifikasi.

3.1.3 Variabel Penelitian

Berikut ini adalah variabel penelitian dan penjelasan dari masing-masing variabel :

- 1) Makanan Ringan Khas Palembang

Tabel 1. Makanan ringan khas Palembang

No	Variabel	Definis Operasi Variabel
1.	Pempek Lenjer	Mempunyai bentuk yang panjang
2.	Pempek Keriting	Mempunyai bentuk seperti otak
3.	Pempek Adaan	Mempunya bentuk bulat seperti bola pimpong
4.	Pempek Kulit	Bentuknya bulat pipih dan warnanya agak gelap
5.	Pempek Tahu	Menggunakan tahu kuning keemasan untuk luaran pempek
6.	Pempek Pistol	Memiliki bentuk lipatan kecil yang baerurutan

7.	Pempek Panggang	Memiliki bentuk bulat pipih yang agak tebal serta terbelah dan mempunyai isian
8.	Pempek Belah	Memiliki bentuk pendek kecil dan terdapat belahan yang ada isinya
9.	Otak-Otak	Ukurannya panjang dan dibungkus dengan daun pisang
10.	Kue Maksubah	Berwarna coklat tanah dan memiliki bercak coklat
11.	Kue Delapan Jam	Memiliki warna coklat tanah
12.	Kue Srikaya	Warnanya hijau daun dan memiliki lubang-lubang kecil di atasnya
13.	Kue Ragit	Mempunya bentuk seperti jala
14.	Kue Gandus	Berwarna putih dan di atasnya ada ebi halus, bawang goreng serta seledri iris
15.	Engkak Ketan	Mempunyai garis coklat yang berlapis-lapis
16.	Lapis Kojo	Berwarna hijau tua pudar dan mempunyai bercak coklat

2) Makanan Sedang Khas Palembang

Tabel 2. Makanan sedang khas Palembang

No	Variabel	Definisi Operasi Variabel
1.	Lenggang	Telur dadar yang terdapat potongan pempek kecil-kecil
2.	Tekwan	Kumpulan pempek kecil-kecil yang berkuah bening
3.	Model Ikan	Kumpulan pempek yang dipotong-potong yang berkuah bening
4.	Model Gandum	Kumpulan pempek berwarna coklat keemasan dipotong-potong yang berkuah bening
5.	Laksan	Kumpulan pempek yang dipotong-potong berkuah kuning keemasan
6.	Celimpungan	Kumpulan pempek bulat tidak beraturan berkuah kuning
7.	Mie Celor	Mie berukuran agak besar warna kuning dan berkuah krem
8.	Burgo	Gulungan berlapis yang dipotong-potong berkuah kuning
9.	Lakso	Mie agak besar berwarna putih dan berkuah kuning
10.	Rujak Mie	Mie agak besar berwarna kuning, berkuah coklat kehitaman dan ada potongan kecil tahu coklat
11.	Pempek Kapal	Ukurannya besar, bagian bawah bulat besar, dan bagian atas

Selam pipih

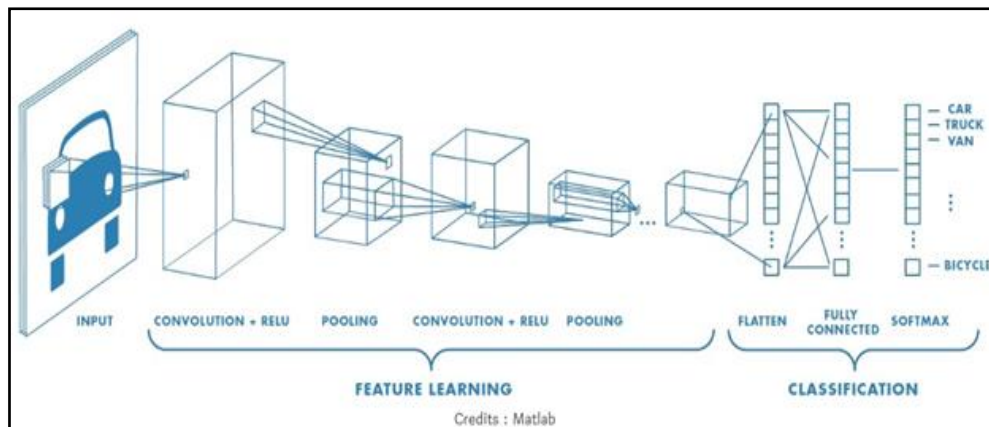
3) Makanan Berat Khas Palembang

Tabel 3. Makanan berat khas Palembang

No	Variabel	Definisi Operasi Variabel
1.	Pindang Paten	Potongan ikan berkuah kuning keemasan, ada daun kemangi hijau tua dan ada cabe merah
2.	Pindang Tulang	Potongan tulang iga berkuah coklat keemasan, ada daun kemangi hijau tua, ada potongan tomat merah, ada cabe merah, dan potongan nanas kuning
3.	Pindang Patin Tempoyak	Potongan ikan berkuah kuning kental
4.	Malbi	Kumpulan potongan dadu daging sapi an berkuah coklat kental

3.2 Metode Analisis Data

Metode Analisis Data yang di gunakan adalah Metode *Convolutional Neural Network (CNN)*.



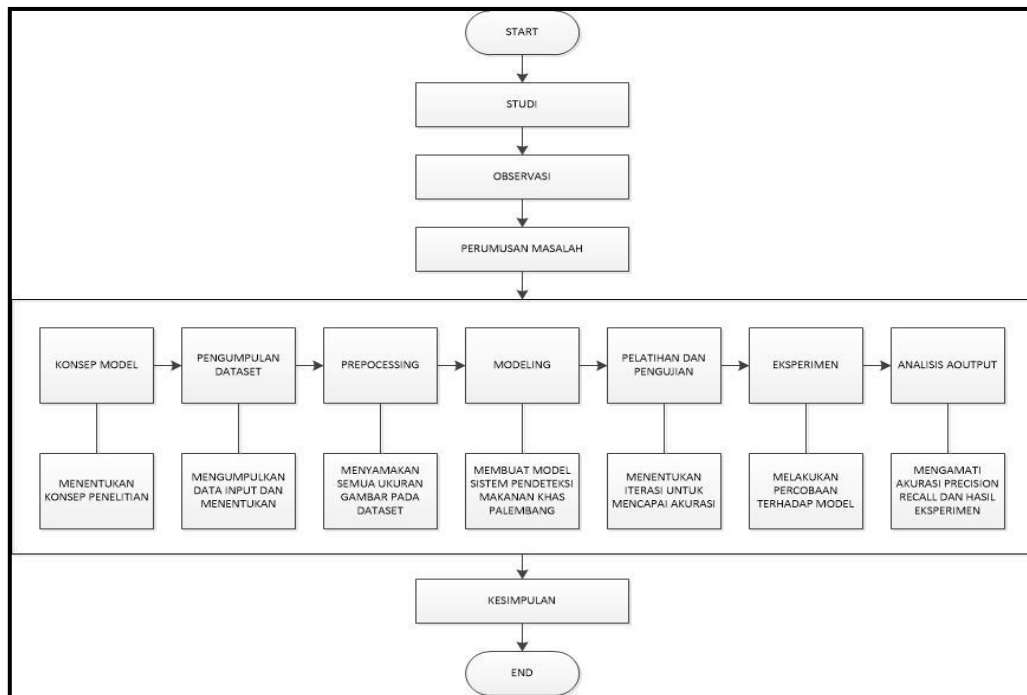
Gambar 1. Arsitektur CNN

Sumber : <https://medium.com>(Sofia, 2018)

Struktur CNN terdiri dari input, proses ekstraksi fitur, proses klasifikasi dan output. Proses ekstraksi dalam CNN terdiri dari beberapa lapisan tersembunyi atau hidden layer, yaitu lapisan konvolusi, fungsi aktivasi (ReLU), dan pooling. CNN bekerja secara hierarki, sehingga output pada lapisan konvolusi pertama digunakan sebagai input pada lapisan konvolusi selanjutnya. Pada proses klasifikasi terdiri dari fully-connected dan fungsi aktivasi (softmax) yang outputnya berupa hasil klasifikasi. Dengan menggunakan metode tersebut peneliti dapat melakukan klasifikasi terhadap gambar makanan khas Palembang yang terdiri dari tiga kelas yaitu makanan ringan khas Palembang, makanan

sedang khas Palembang, dan makanan berat khas Palembang.

3.3 Tahapan Penelitian



Gambar 2. Kerangka Berfikir

4. Hasil dan Pembahasan

4.1 Pengumpulan Dataset Data Citra Makanan Khas Palembang

Tahap pertama yang dilakukan penulis dalam merancang sistem adalah dengan mengumpulkan dataset makanan khas Palembang, yang akan digunakan untuk data train dan data uji. Pengumpulan dataset didapatkan dari kumpulan data gambar terbuka Google Image, terdapat 31 jenis makanan khas Palembang, masing-masing jenis terdiri dari kurang lebih 50 sampai 70 gambar, jadi total gambar yang digunakan dari 31 jenis makanan Palembang yaitu 1955 gambar dengan format jpg. Gambar di simpan ke sebuah folder berdasarkan jenis maakanan Khas Palembang.

4.2 Modifikasi Dataset Data Citra Makanan Khas Palembang

Setelah data dikumpulkan, data di edit kembali menggunakan adobe Photoshop CS6 yang bertujuan untuk mencrop data dan mengubah ukuran gambar yang bertujuan untuk mendapatkan data baru dengan ukuran gambar yang sama.

4.3 Labelimg

Setelah data selesai dimodifikasi, beri label pada kotak pembatas objek dalam gambar menggunakan *LabelImg* oleh *tzutalin*. *LabelImg* adalah alat anotasi gambar grafis dan kotak pembatas objek label dalam gambar. Sebelum menggunakan aplikasi *LabelImg*, perbaruhi file `predefined_classes.txt` yang terdapat pada folder data, edit file tersebut

seperti dibawah ini :

```

burgo
celimpungan
malbi
pindang patin
pindang tulang
tempoyak patin
laksan
lakso
lenggang
mie celor
model gandum
model ikan
rujak mie
tekwan
delapan jam
enggak ketan
gandus
lapis kojo
srikaya
otak otak
pepek tahu
pepek kulit krispi
pepek telur
pepek campur
pepek panggang
pepek lenjer
pepek pistel
pepek kilit
pepek keriting
pepek kapal selam
pepek belah
pepek adaan
ragit
maksuba

```

Gambar 3. Isi file predefined_classes.txt

Setelah itu labeling gambar satu persatu, simpan gambar dan file anotasi ke folder img setelah itu compress folder img ke format .zip. Pemberian label pada gambar ini bertujuan untuk mendapatkan ciri khusus yang merepresentasikan masing-masing objek yang terdapat dalam gambar. Ciri khusus tersebut nantinya akan dijadikan learning dalam proses training sekaligus digunakan pada tahap pengujian. Hasil labeling gambar tersebut berupa file txt yang mereperesentasikan koordinat objek / kotak pembatas dan nama kelas / id kelas dari objek tersebut (nilai skala dari 0 hingga 1). Karena setiap gambar hanya berisi 1 objek, maka cukup menggambar 1 kotak pembatas sehingga membuat file teks hanya berisi 1 baris.

```

0 0.508654 0.755072 0.594231 0.489855

```

Gambar 4. Hasil labeling

Setiap baris ada 5 kolom, id kelas dan kotak pembatas dengan format (id_kelas, x, y, lebar, tinggi) : <class> <x> <y> <width> <height>

4.4 Pelatihan dan Pengujian

Untuk memulai pelatihan menggunakan *Google Colab*, buat file *Google Colab* dan konfigurasi *Runtime* untuk bekerja dengan GPU. GPU di aktifkan agar *Notebook Colab* dalam menjalankan sistem YOLO dapat memproses deteksi lebih dari 100 lebih cepat dari CPU. import beberapa perpustakaan berikut yang akan diperlukan saat pengujian

```

import cv2
import matplotlib.pyplot as plt
import glob

```

```
from google.colab import drive
```

Setelah itu hubungkan *Notebook Colab* ke *Google Drive* dengan perintah sebagai berikut :

```
drive.mount('/content/gdrive')
!ln -s /content/gdrive/my\ drive/ / mydrive
!ls /mydrive
```

Kloning dan bangun darknet menggunakan perintah dibawah ini :

```
!git clone https://github.com/AlexeyAB/darknet.git
%cd darknet
!sed -i 's/OPENCV=0/OPENCV=1/' Makefile
!sed -i 's/GPU=0/GPU=1/' Makefile
!sed -i 's/CUDNN=0/CUDNN=1/' Makefile
!make
```

Buat salinan dari file `yolov3.cfg` yang ada di darknet dan ganti nama file menjadi `yolov3_custom.cfg` dengan perintah dibawah ini :

```
!cp cfg/yolov3.cfg cfg/yolov3_custom.cfg
```

Setelah itu perbarui file `yolov3_custom.cfg` dengan mengubah batch, subdivisi, `max_batches`, kelas, dan filter dengan perintah dibawah ini :

```
# Change classes value to your number of objects
classes = 34
filters = (classes + 5) * 3
max_batches = classes * 2000

if max_batches < 6000:
    max_batches = 6000

# Edit classes & filters
!sed -i 's/batch=1/batch=64/'
cfg/yolov3_custom.cfg
!sed -i 's/subdivisions=1/subdivisions=16/'
cfg/yolov3_custom.cfg
!sed -i 's/max_batches = 500200/max_batches = $max_batches/'
cfg/yolov3_custom.cfg
!sed -i '610 s@classes=80@classes=$classes@'
cfg/yolov3_custom.cfg
!sed -i '696 s@classes=80@classes=$classes@'
cfg/yolov3_custom.cfg
!sed -i '783 s@classes=80@classes=$classes@'
cfg/yolov3_custom.cfg
!sed -i '603 s@filters=255@filters=$filters@'
cfg/yolov3_custom.cfg
!sed -i '689 s@filters=255@filters=$filters@'
cfg/yolov3_custom.cfg
!sed -i '776 s@filters=255@filters=$filters@'
cfg/yolov3_custom.cfg
```

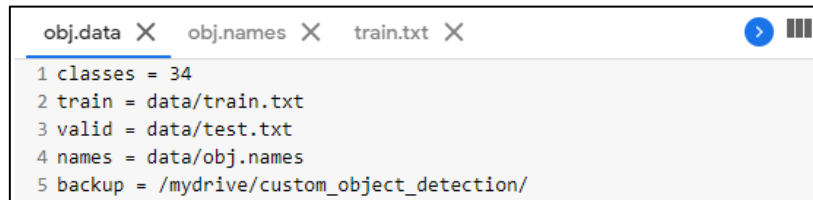
setelah itu upload dataset img.zip ke folder custom_object_detection di Google Drive kemudian buat direktori baru pada darknet dengan nama data/obj dan ekstrak img.zip ke dalam folder tersebut dengan perintah sebagai berikut:

```
!mkdir data/obj
!unzip /mydrive/custom_object_detection/img.zip -d data/obj
```

Buat file obj.data dengan perintah dibawah ini :

```
!echo -e 'classes = $classes\ntrain = data/train.txt\nvalid = data/test.txt\nnames = data/obj.names\nbackup = /mydrive/custom_object_detection/' > data/obj.data
```

Berikut merupakan hasil dari perintah diatas:



```
obj.data X obj.names X train.txt X
1 classes = 34
2 train = data/train.txt
3 valid = data/test.txt
4 names = data/obj.names
5 backup = /mydrive/custom_object_detection/
```

Gambar 5. File obj.data

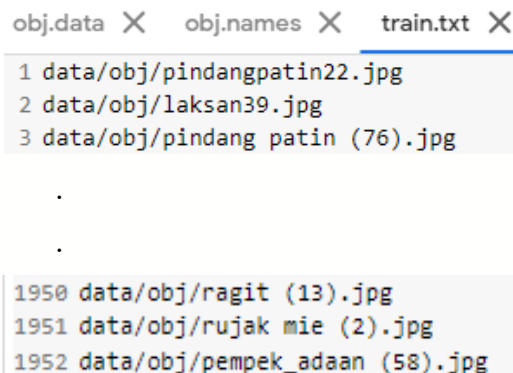
Buat file baru bernama obj.names. Isi file dari obj.names sama persis dengan file predefined_classes.txt (saat membuat label gambar) kemudian upload file tersebut didalam googledrive, setelah itu copy paste file ke folder data di darknet dengan cara sebagai berikut :

```
!cp /mydrive/custom_object_detection/obj.names data
```

Ambil dataset gambar dari direktori data/obj/ dan taruh di data/train.txt

```
images_list = glob.glob("data/obj/*[jpg|png|jpeg]")
print(images_list)
# Create train.txt file
file = open("data/train.txt", "w")
file.write("\n".join(images_list))
file.close()
```

Berikut merupakan hasil dari perintah diatas:



```
obj.data X obj.names X train.txt X
1 data/obj/pindangpatin22.jpg
2 data/obj/laksan39.jpg
3 data/obj/pindang patin (76).jpg
.
.
1950 data/obj/ragit (13).jpg
1951 data/obj/rujak mie (2).jpg
1952 data/obj/pempek_adan (58).jpg
```

Gambar 6. File train.txt

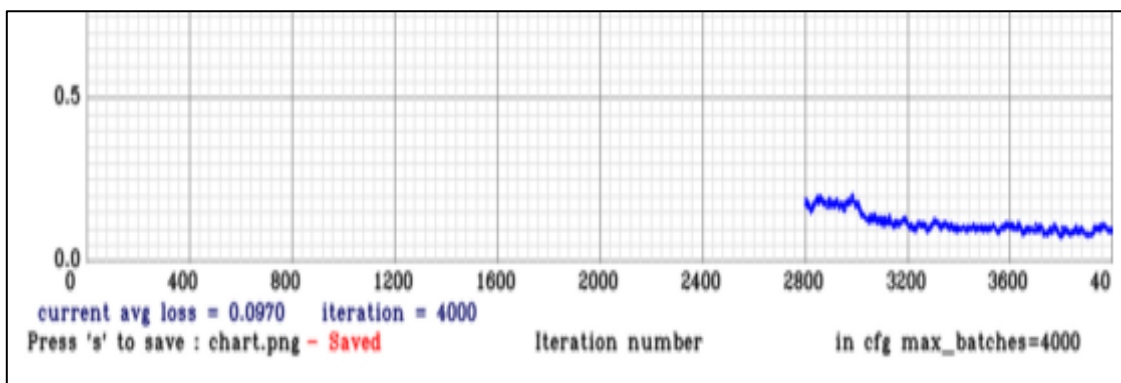
Unduh bobot yang telah dilatih sebelumnya menggunakan perintah wget

```
!wget https://pjreddie.com/media/files/darknet53.conv.74
```

Proses awal training untuk data pelatihan sebanyak 1995 gambar dengan menjalankan perintah dibawah ini :

```
!./darknet detector train data/obj.data cfg/yolov3_custom.cfg darknet53.conv.74 -  
dont_show
```

Konfigurasi training pada pelatihan ini mencapai 4000 iterasi dengan avg loss 0,0970. Proses konfigurasi training akan disimpan secara bertahap pada iterasi per-kelipatan 1000. Iterasi terakhir akan tersimpan dengan nama yolov3-custom_last.weights yang akan digunakan sebagai detektor objek. Proses ini akan disimpan di folder backup.



Gambar 7. Grafik Loss pada iterasi 4000

Sebelum melakukan pengujian, buat folder bernama dataset_uji didalam folder custom_object_detection di google drive, simpan gambar yang akan di uji di folder tersebut. Berikut merupakan sel untuk menjalankan detektor objek kustom :

```
!./darknet detector test data/obj.data cfg/yolov3_custom.cfg /mydrive/custom_objec  
t_detection/yolov3_custom_last.weights /mydrive/custom_object_detection/dataset  
_uji/pempeklej1.jpg -thresh 0.25
```

Hasil deteksi akan disimpan dengan nama predictions.jpg

```
pred = cv2.imread('predictions.jpg')  
pred = cv2.cvtColor(pred, cv2.COLOR_BGR2RGB)  
plt.figure(figsize=(20,20))  
plt.imshow(pred)
```

Berikut merupakan hasil dari pengujian :

```
[yolo] params: iou loss: mse (2), iou_norm: 0.75, obj_norm: 1.00, cls_norm: 1.00, delta_norm: 1.00, scale_x_y: 1.00
Total BFLOPS 65.544
avg_outputs = 523290
Allocate additional workspace_size = 52.43 MB
Loading weights from /mydrive/custom_object_detection/yolov3_custom_last.weights...
seen 64, trained: 256 K-images (4 Kilo-batches_64)
Done! Loaded 107 layers from weights-file
Detection layer: 82 - type = 28
Detection layer: 94 - type = 28
Detection layer: 106 - type = 28
/mydrive/custom_object_detection/dataset_uji/pempeklenjer1.jpg: Predicted in 40.363000 milli-seconds.
pempek lenjer: 92%
Unable to init server: Could not connect: Connection refused

(predictions:1223): Gtk-WARNING **: 03:24:49.325: cannot open display:
```

Gambar 8. Output Pengujian

Didapatkan hasil bahwa gambar uji dari pempek lenjer mendapat akurasi sebesar 92% dengan waktu prediksi 40,363000 milli-second.



Gambar 9. Hasil Pengujian

Proses pengujian dilakukan dengan memasukan gambar yang diambil secara acak di google images dengan posisi objek yang terlihat jelas. gambar disimpan di folder dataset_uji pada google drive, kemudian dilakukan deteksi dengan memanggil path gambar pada source code detector test. Dari deteksi yang telah dilakukan didapatkan hasil akurasi dan kecepatan deteksi pada 31 makanan Khas Palembang seperti yang terlihat pada ketiga tabel diatas dengan hasil pengujian menampilkan kotak jangkar dan nama label pada gambar. Berikut merupakan rangkuman hasil akurasi dan kecepatan deteksi :

Tabel 4. Rangkuman Akurasi Hasil Pengujian

No	Variabel	Akurasi	Kecepatan Deteksi
1.	Pempek Lenjer	92%	40.363000 milli-seconds
2.	Pempek Keriting	96%	40.438000 milli-seconds.

3.	Pempek Adaan	96%	40.483000 milli-seconds.
4.	Pempek Kulit	99%	40.378000 milli-seconds.
5.	Pempek Tahu	90%	40.484000 milli-seconds
6.	Pempek Pistel	100%	40.417000 milli-seconds.
7.	Pempek Panggang	98%	40.448000 milli-seconds
8.	Pempek Belah	94%	40.450000 milli-seconds
9.	Otak-Otak	89%	40.572000 milli-second
10.	Kue Maksubah	100%	40.631000 milli-seconds.
11.	Kue Delapan Jam	99%	40.451000 milli-seconds.
12.	Kue Srikaya	90%	40.577000 milli-seconds.
13.	Kue Ragit	100%	40.612000 milli-seconds.
14.	Kue Gandus	99%	40.586000 milli-seconds.
15.	Engkak Ketan	100%	41.101000 milli-seconds.
16.	Lapis Kojo	84%	40.478000 milli-seconds.
17.	Lenggang	99%	40.433000 milli-seconds.
18.	Tekwan	100%	40.457000 milli-seconds
19.	Model Ikan	100%	40.480000 milli-seconds.
20.	Model Gandum	97%	40.579000 milli-seconds.
21.	Laksan	97%	40.440000 milli-seconds.
22.	Celimpungan	88%	40.418000 milli-seconds.
23.	Mie Celor	87%	40.424000 milli-seconds
24.	Burgo	100%	40.413000 milli-seconds.
25.	Lakso	100%	40.416000 milli-seconds.
26.	Rujak Mie	95%	40.417000 milli-seconds.
27.	Pempek Kapal Selam	95%	40.460000 milli-seconds.
28.	Pindang Paten	97%	40.386000 milli-seconds.
29.	Pindang Tulang	100%	40.463000 milli-seconds.

30.	Pindang Patin Tempoyak	95%	40.428000 milli-seconds.
31.	Malbi	99%	40.387000 milli-seconds
Rata-rata :		96%	40.486.129 milli-seconds

Berdasarkan tabel Rangkuman Akurasi Hasil Pengujian diatas didapatkan bahwa, nilai akurasi terkecil merupakan 84% sedangkan nilai akurasi terbesar merupakan 100%. Kecepatan deteksi terendah adalah 41.101000 milli-seconds sedangkan kecepatan deteksi tertinggi adalah 41.101000 milli-seconds. Dari 31 variabel makanan khas Palembang yang telah di uji didapat nilai rata-rata akurasi sebesar 96% dan rata-rata kecepatan deteksi sebesar 40.486.129 milli-seconds. Maka dapat disimpulkan bahwa tingkat keakurasian model algoritma YOLOv3 yang digunakan dikatakan baik dan cepat, karena hasil rata-rata akurasi diatas 80%.

5. Kesimpulan

Setelah dilakukan pengumpulan dataset citra makanan khas Palembang, modifikasi dataset, anotasi gambar grafis dan kotak pembatas objek label dalam gambar dataset, pelatihan dan pengujian terhadap perancangan sistem yang dilakukan menggunakan google colab yang menyediakan GPU hingga 12GB sehingga saat menjalankan sistem YOLOv3 dapat memproses deteksi 100 kali lebih cepat dari CPU, sehingga mempermudah dan mempercepat perancangan sistem deteksi 31 makanan khas Palembang. Maka eksperimen yang dilakukan terhadap 31 variabel penelitian menghasilkan rata-rata akurasi sebesar 96% dan rata-rata kecepatan deteksi sebesar 40.486.129 milli-seconds. Maka dapat disimpulkan bahwa tingkat keakurasian model algoritma YOLOv3 yang digunakan dikatakan baik dan cepat, karena hasil rata-rata akurasi diatas 80%.

Referensi

- Antoni, D., Herdiansyah, M. I., Akbar, M., & Sumitro, A. (2021). Pengembangan Infrastruktur Jaringan Untuk Meningkatkan Pelayanan Publik di Kota Palembang. *JURNAL MEDIA INFORMATIKA BUDIDARMA*, 5(4), 1652-1659.
- Antoni, D., Jie, F., & Abareshi, A. (2020). Critical factors in information technology capability for enhancing firm's environmental performance: case of Indonesian ICT sector. *International Journal of Agile Systems and Management*, 13(2), 159-181.
- Antoni, D., & Akbar, M. (2019). E-supply chain management value concept for the palm oil industry. *Jurnal Sistem Informasi*, 15(2), 15-29.
- Antoni, D., Fikari, D., & Akbar, M. (2018). The readiness of palm oil industry in enterprise resource planning. *Telkomnika*, 16(6), 2692-2702.
- Fauzi, F., Dencik, A. B., & Asiati, D. I. (2019). Metodologi Penelitian untuk manajemen dan akuntansi. *Jakarta: Salemba Empat*.
- Karlina, o. E., & indarti, d. (2020). Pengenalan objek makana cepat saji pada video dan real time webcam menggunakan metode you look only once (yolo). *Jurnal ilmiah informatika komputer*, 24(3), 199–

208. <https://doi.org/10.35760/ik.2019.v24i3.2362>
Kartika, t., & harahap, z. (n.d.). *The culinary development of pempek as a gastronomic tourist attraction in Palembang, Sumatera Selatan*. 24.
- Sri, j. w. h. (2019). deteksi kendaraan secara real-time menggunakan metode yolo berbasis android. *deteksi kendaraan secara real-time menggunakan metode yolo berbasis android*.
<https://informatika.ft.umrah.ac.id/>
- rohcastu, t. k., rahmad, c., & rawansyah. (2019). object detection system sebagai alat bantu mendeteksi objek sekitar untuk penyandang tunanetra. *seminar informatika aplikatif polinema*, 81–88.
- Adam. R., R. 2016. Deep Learning Untuk Pengenalan Pelafalan Huruf Hijaiyah Berharakat. UGM: Program Studi Ilmu Komputer. Skripsi.
- American Society of Media Photographers. 2016. *Color Space and Color Profile*. Diperoleh pada 12 Februari 2018
<http://www.dpbestflow.org/color/color-space-and-color-profiles>.
- Arrofiqoh, E. N., & Harintaka, H. (2018). IMPLEMENTASI METODE CONVOLUTIONAL NEURAL NETWORK UNTUK KLASIFIKASI TANAMAN PADA CITRA RESOLUSI TINGGI. *GEOMATIKA*, 24(2), 61. <https://doi.org/10.24895/JIG.2018.24-2.810>
- Bakri, M. (2017). *KLASIFIKASI MOTIF KAIN TRADISIONAL BATIK BOMBA KAILI BERDASARKAN FITUR TEKSTUR CITRA DIGITAL*. 9.
- Bronlee, J. 2017. *Gentle Introduction to the Adam Optimization Algorithm for Deep Learning*. Diperoleh pada 12 Februari 2018
<https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/>
- Djumena, Nian S.1990a. *Ungkapan Sehelai Batik: It's Mystery and Meaning*. Jakarta: Djabatan.
- Deng, L. dan Yu, D. 2014. Deep Learning: Methods and Applications, *Foundations and Trends in Signal Processing*. 7. 3-4. 197-387.
- Fausett, L. 1994. *Fundamentals of Neural Networks: Achitectures, Algorithms, and Applications*. New Jersey: Prentice Hall.
- Gonzalez, R.C. dan Woods, R.E. 2008. *Digital Image Processing 3rd ed*. Prentice

Hall.

Goodfellow, I., Bengio, Y., dan Courville, A.. 2016. *Deep Learning*. Diperoleh pada 25 Januari 2018 dari <http://goodfeli.github.io/dlbook/>.

Copyrights

Copyright for this article is retained by the author(s), with first publication rights granted to the journal.

This is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>)