

PERANCANGAN APLIKASI PERHITUNGAN NUTRISI PADA MAKANAN BERBASIS ANDROID DENGAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN)

THE DESIGN OF NUTRITION CALCULATION APPLICATION FOR ANDROID USING CONVOLUTIONAL NEURAL NETWORK (CNN) METHOD

Mochamad Dandi¹, Hilman Fauzi TSP², Syamsul Rizal³

^{1,2,3} Universitas Telkom, Bandung

¹mchmddandi@student.telkomuniversity.ac.id, ²hilmanfauzitsp@telkomuniversity.ac.id,

³syamsul@telkomuniversity.ac.id

Abstrak

Pemenuhan kebutuhan nutrisi dari segi jumlah dan jenisnya sangat penting dalam membantu proses tumbuh kembang manusia. Upaya peningkatan kualitas sumber daya manusia (SDM) dimulai dari pemenuhan kebutuhan dasar manusia. Perhatian utamanya terletak pada proses tumbuh kembang anak sejak pembuahan sampai mencapai dewasa muda. Mengonsumsi makanan yang bergizi merupakan salah satu faktor penting untuk menjaga gaya hidup yang sehat. Untuk mendapatkan makanan yang bergizi diperlukan komposisi yang tepat. Untuk mendapatkan jumlah kalori yang tepat diperlukan pengukuran, dalam mengukur jumlah kalori dalam makanan terdapat dua metode yaitu dengan metode konvensional dan metode *image processing*. Metode konvensional bisa menggunakan perhitungan *Base Metabolic Rate* (BMR) dengan jumlah kalori yang dibutuhkan tubuh seseorang untuk aktifitas sehari-hari serta jumlah kalori makanan yang dikonsumsi. Sedangkan penggunaan metode *image processing* belum banyak diterapkan pada masyarakat Indonesia. Maka dalam Tugas Akhir ini dilakukan simulasi Perancangan Aplikasi perhitungan nutrisi berbasis android, dan simulasi ini dilakukan menggunakan metode *Convolutional Neural Network* (CNN). Data yang dipakai di metode ini adalah gambar makanan yang dipakai untuk metode penelitian. Akan diukur hasil performansinya untuk membuktikan performa aplikasi perhitungan. Pengerjaan simulasi Tugas Akhir ini yaitu dapat membuktikan performa aplikasi perhitungan nutrisi pada makanan berbasis android, dengan melakukan pengukuran dari sisi akurasi data, *image processing*.

Keyword: ImageProcessing Convolutional Neural Network, Android

Abstract

Fulfilling nutritional needs in terms of quantity and type is very important in helping the process of human growth and development. Efforts to improve the quality of human resources (HR) start from meeting basic human needs. Its main concern lies in the development process of the child from conception to young adulthood. Consuming nutritious food is one of the important factors for maintaining a healthy lifestyle. To get nutritious food, you need the right composition. Right number of calories, you need measurement. There are two methods to measure the number of calories in food, namely the conventional method and the image processing method. Conventional methods can use the calculation of the Base Metabolic Rate (BMR) with the number of calories a person's body needs for daily activities and the number of food calories consumed. The use of image processing methods has not been widely applied in Indonesian. The data used in this method is a picture of the food for the research method. Then the performance results will be measured to prove the performance the application. This Final Project simulation work to prove the performance application of nutrient calculation on android, by measuring in terms of data accuracy, image processing.

Keyword: ImageProcessing Convolutional Neural Network, Android

1. Pendahuluan

Menurut Undang-Undang Republik Indonesia Nomor 36 tahun 2009 pasal 1 Kesehatan adalah keadaan sehat, baik secara fisik, mental, spritual maupun sosial yang memungkinkan setiap orang untuk hidup produktif secara sosial dan ekonomis. Oleh karena kesehatan adalah hal terpenting dari kehidupan. Untuk menjaga kesehatan tubuh, memerlukan pola gaya hidup yang baik diantaranya seperti istirahat yang cukup, olahraga yang teratur, serta mengonsumsi makanan yang sehat dan serta kaya akan nutrisi dan gizi[1].

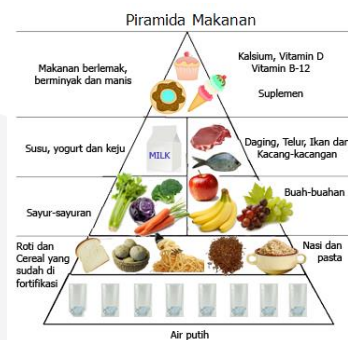
Nutrisi adalah proses dimana tubuh manusia menggunakan makanan untuk membentuk energi, mempertahankan kesehatan, pertumbuhan dan untuk berlangsungnya fungsi normal setiap organ baik antara asupan nutrisi dengan kebutuhan nutrisi [2]. Sebelum membahas lebih dalam tentang nutrisi maka kita harus terlebih dahulu tentang gizi. Gizi menurut kamus besar Bahasa Indonesia adalah zat makanan pokok yang diperlukan bagi pertumbuhan dan kesehatan badan[1]. Seperti kita ketahui bahwa zat gizi terkandung dalam bahan makanan berupa karbohidrat, protein, lemak, vitamin, mineral dan air. Keenam unsur ini di golongkan menjadi 3 kelompok yaitu, yang pertama unsur gizi dimana yang termasuk didalamnya adalah karbohidrat, protein dan lemak. Yang kedua unsur gizi pembangun dimana di unsur gizi kedua ini protein, mineral dan air termasuk pada unsur gizi pembangun lalu yang terakhir adalah unsur gizi yang membantu mengatur fungsi alat-alat tubuh, termasuk dalam kelompok ini adalah vitamin.[1]. Mengonsumsi makanan yang bergizi merupakan salah satu faktor penting untuk menjaga gaya hidup

yang sehat. Untuk mendapatkan makanan yang bergizi di perlukan komposisi yang tepat. Formula makanan yang di anjurkan untuk memenuhi kebutuhan sehari hari yaitu 2.100 kkal, dengan kandungan protein 07-12 persen, lemak 35-45 persen dari total energi [3]. Formula itu disusun dengan menggunakan Daftar Komposisi Pangan Indonesia [4]. Untuk mendapat jumlah kalori, gizi dan nutrisi yang tepat di perlukan pengukuran, dalam mengukur jumlah kalori, gizi dan nutrisi dalam makanan terdapat dua metode yaitu dengan metode konvensional dan metode *image processing*. Metode konvensional bisa menggunakan perhitungan dengan jumlah kalori yang dibutuhkan tubuh seseorang untuk aktifitas serta jumlah kalori makanan yang dikonsumsi [6]. Sedangkan metode dengan *image processing* dilakukan dengan mengenali objek yang akan diukur selanjutnya mengukur volume objek dan bentuk kontur objek [7]. Penelitian-penelitian sebelumnya lebih berfokus pada ekstraksi ciri untuk kalori makanan berkuah dengan menggunakan metode *Support Vector Machine* (SVM) tetapi tidak berbasis android. Di penelitian ini akan menggunakan metode *Convolutional Neural Network* dengan basis android dimana data akan lebih akurat karena proses *Convolutional Neural Network* membutuhkan data yang lebih banyak dibandingkan data *Support Vector Machine*. Pada saat ini perhitungan nutrisi pada makanan masih menggunakan metode konvensional sehingga perlu di ciptakannya metode yang lebih praktis untuk di lakukan pengukuran yaitu dengan menggunakan metode *image processing* yang akan diterapkan pada aplikasi berbasis android. Diharapkan aplikasi ini dapat membantu untuk mengontrol jumlah nutrisi dalam makanan.

2. Konsep Dasar

2.1 Nutrisi

Sebelum membahas dasar nutrisi, maka perlu mengetahui tentang istilah kalori, gizi dan nutrisi. Kalori merupakan satuan unit yang di gunakan untuk menghitung nilai energi pada tubuh ketika mengkonsumsi makanan. Berbeda dengan gizi, gizi menurut kamu besar Bahasa Indonesia adalah zat makanan pokok yang di perlukan bagi pertumbuhan dan kesehatan badan. Sedangkan nutrisi adalah makanan bergizi atau proses pemasukan dan pengolahan zat makanan oleh tubuh[5].



Gambar 2. 1 Piramida Makanan

2.2 Convolutional Neural Network

Convolutional Neural Network adalah salah satu metode *machine learning* dari pengembangan *Multi Layer Perceptron* (MLP) yang didesain untuk mengolah data dua dimensi. CNN termasuk dalam jenis *Deep Neural Network* karena dalamnya tingkat jaringan dan banyak diimplementasikan dalam data citra. CNN memiliki dua metode; yakni klasifikasi menggunakan *feedforward* dan tahap pembelajaran menggunakan *backpropagation*. Cara kerja CNN memiliki kesamaan pada MLP, namun dalam CNN setiap neuron dipresentasikan dalam bentuk dua dimensi, tidak seperti MLP yang setiap neuron hanya berukuran satu dimensi. *Convolutional Neural Network* memiliki beberapa layer yang difungsikan untuk melakukan filter pada setiap prosesnya. Prosesnya disebut dengan proses training. Pada proses training terdapat 3 tahapan yaitu *Convolutional layer*, *Pooling layer*, dan *Fully connected layer*[9].

2.3 Image Processing

Image Processing adalah suatu metode untuk mengolah gambar (Image) ke dalam bentuk digital untuk tujuan tertentu. Pada awalnya pengolahan citra ini berfungsi untuk memperbaiki dan meningkatkan kualitas suatu gambar, namun dengan perkembangan zaman dan munculnya ilmu-ilmu komputasi hal itu memungkinkan manusia dapat mengambil suatu informasi yang ada dalam suatu gambar. Inputnya adalah citra (gambar) dan keluarannya adalah citra yang sudah diperbagus kualitasnya. Misalnya, sebuah gambar yang kurang tajam warnanya, kabur (blurring) dan terdapat noise (mis bintang putih) memerlukan proses untuk memperbaiki kualitas gambar sehingga mendapatkan informasi yang lebih baik[9].



Gambar 2.2 Image Processing

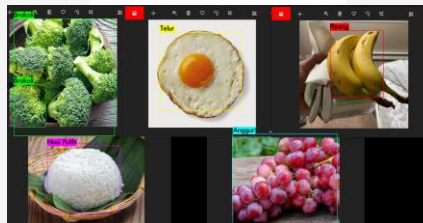
2.4 Android Studio

Android Studio adalah Lingkungan Pengembangan Terpadu – Integrated Development Environment (IDE) untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA. Selain merupakan editor code IntelliJ dan alat pengembang yang berdaya guna, Android Studio menawarkan lebih banyak fitur. Gunanya untuk meningkatkan produktivitas kamu saat membuat aplikasi Android, misalnya[10]:

1. Sistem versi berbasis Gradle yang fleksibel
2. Emulator yang cepat dan kaya fitur
3. Lingkungan yang menyatu untuk pengembangan bagi semua perangkat Android
4. Instant Run untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru
5. Template kode dan integrasi GitHub untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh
6. Alat pengujian dan kerangka kerja yang ekstensif
7. Alat Lint untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan masalah-masalah lain.
8. Dukungan C++ dan NDK.

2.5 Object Detection

Object Detection adalah teknologi komputer yang terkait dengan visi komputer dan pemrosesan gambar yang berhubungan dengan pendeteksian objek semantik dari kelas tertentu (seperti manusia, bangunan, atau mobil) dalam gambar dan video digital[11].



Gambar 2.3 Object Detection

2.6 You Only Look Once (YOLO)

YOLO merupakan salah satu state-of-the-art object detection algorithm yang membagi citra masukan ke dalam suatu grid berukuran $S \times S$. Ukuran dari grid cell tersebut tergantung pada input size yang digunakan pada suatu arsitektur. Pada YOLOv3, jika input size 416x416, maka ukuran grid size adalah 13x13, 26x26, dan 52x52. Setiap sel bertugas untuk memprediksi objek yang ada di dalam sel tersebut dengan bounding box beserta confidence yang merupakan nilai probabilitas keberadaan suatu objek pada bounding box tersebut. Kemudian, setelah bounding box dipetakan berdasarkan nilai confidence yang dihasilkan, YOLO akan memprediksi kelas dari objek yang terdapat pada bounding box tersebut beserta probabilitasnya, sehingga terbentuklah class probability map. Dari sekian banyak bounding box yang dihasilkan, untuk mendapatkan bounding box beserta kelas objek dengan probabilitas yang tinggi, maka dari seluruh hasil prediksi tersebut, hanya yang melampaui threshold saja yang akan digunakan. Jika terdapat duplikasi bounding box, maka Non-max Suppression (NMS) berperan untuk menghilangkan duplikat tersebut. Gambar 1 merupakan ilustrasi dari cara kerja YOLO[12].

Pada YOLOv3, feature extractor yang sebelumnya menggunakan Darknet-19 menjadi Darknet-53, dan juga proses deteksi objek yang kini menggunakan 3 skala di mana pada setiap skala menggunakan 3 anchor boxes sehingga berdampak pada meningkatnya kemampuan mendeteksi objek yang lebih kecil[12].



Gambar 2.4 YOLOv3 Object Detection[12]

2.7 Application Program Interface (API)

API atau Application Programming Interface adalah sebuah interface yang dapat menghubungkan aplikasi satu dengan aplikasi lain. API berperan sebagai perantara antar berbagai aplikasi berbeda, baik dalam satu platform yang sama atau lintas platform. Dalam pengembangan API diperlukan pengenalan arsitektur sebagai pedoman cara menghubungkan antara logic data dengan logic interface. Salah satu gaya arsitektur mengembangkan API adalah berbasis web yang digunakan oleh HTTP dalam berkomunikasi yaitu *Representational State Transfer*[13].

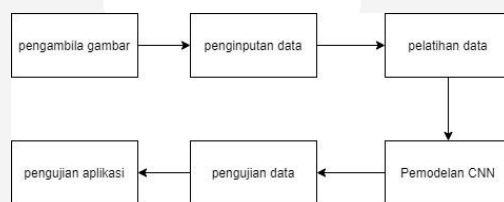


Gambar 2.5 Application Program Interface (API).

3. Model dan sistem perancangan

3.1 Desain Sistem

Pada perancangan tugas akhir dibutuhkan *flowchart*, yaitu tahapan dalam melaksanakan perancangan yang ditunjukkan pada Gambar 3.1. Tujuan dari *flowchart* yaitu agar pelaksanaan tugas akhir dapat bekerja secara terstruktur. Tahap awal dari perancangan adalah pengkondisian dataset yang digunakan dalam tugas akhir. Kemudian Setelah diperoleh semuda data, selanjutnya masuk ketahapan pelatihan *convolutional neural network* lalu melakukan model untuk *convolutional neural network*.



Gambar 3. 1 Diagram Blok Deskripsi Sistem

Jika permodelan simulasi *convolutional neural network* telah sesuai dengan spesifikasi, maka akan dilanjutkan ke dalam tahap konfigurasi hasil model *convolutional neural network* dengan Aplikasi Android selanjutnya maka akan dilakukan analisis. Tahapan terakhir adalah membuat kesimpulan dari perancangan tugas akhir.

3.2 Fungsi dan Fitur Perhitungan Nutrisi

Perhitungan Nutrisi ini memiliki fitur dan fungsi yang sama dengan latar belakang dan tujuan penelitian sebagai berikut:

1. Mampu menghitung kadar nutrisi pada objek yang dideteksi
2. Mampu mengetahui objek apa yang dideteksi
3. Mampu mengetahui objek secara langsung dan tidak

3.3 Perancangan Sistem

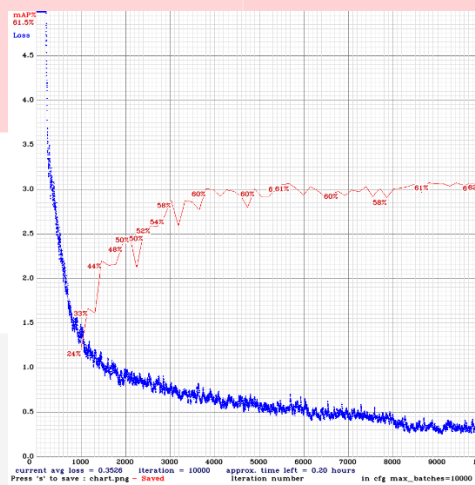
Pada Perancangan system ini akan menjelaskan tentang konfigurasi aplikasi yang akan dibuat. Tujuannya adalah untuk menguji perangkat sistem aplikasi yang dibuat pada Tugas Akhir ini dan menguji akurasi dataset perancangan aplikasi. Konfigurasi dan perancangan ini meliputi:

1. Pengkondisian Dataset
2. Konfigurasi YOLOv3
3. Pelatihan YOLOv3
4. Perancangan Aplikasi Android
5. Perancangan API

4. Hasil Pengujian dan Analisis

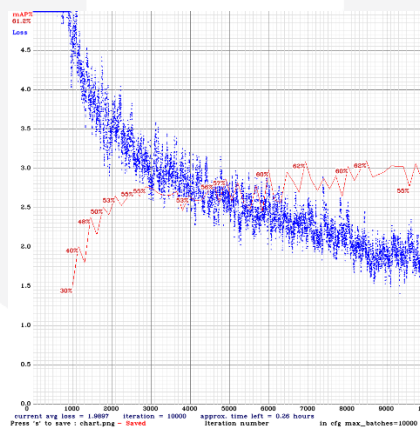
4.1 Pengujian Skenario Pertama

Pengujian dilakukan untuk mengetahui mana yang lebih stabil pada system. System harus mempunyai avg loss dan mAP yang baik pada saat pelatihan model YOLO. Pada Skenario Pertama pengujian dilakukan perbandingan antara YOLOv3 dan YOLOv4 mana yang lebih stabil pada kedua algoritma ini. Dengan menggunakan data uji dan data latih yang sama, jumlah dataset yang sama. Ini dilakukan untuk memilih yang lebih baik untuk digunakan pada pengaplikasian di android. Hasil ditunjukkan pada gambar



Gambar 4. 1 Hasil Pengujian YOLOv3

Berdasarkan Gambar 4.1 dapat disimpulkan bahwa model YOLOv3 mempunyai avg loss yang cukup rendah yaitu 0.3526 dan mAP yang cukup baik menyentuh angka 61.5%. angka ini dinilai baik dengan dataset dan object yang sudah disiapkan artinya maka di model YOLOv3 ini tingkat object detection mengenali object cenderung tinggi.



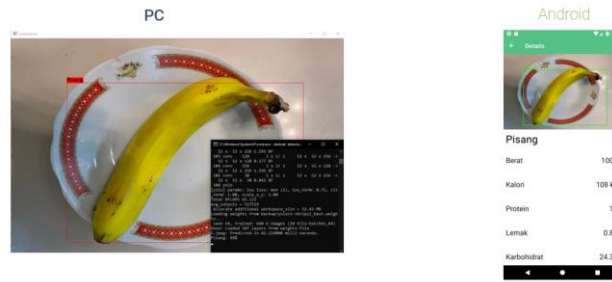
Gambar 4. 2 Hasil Pengujian YOLOv4

Berdasarkan Gambar 4.2 dapat disimpulkan bahwa model YOLOv4 mempunyai avg loss yang cukup tinggi yaitu 1.9897 dan mAP menyentuh angka 61.2%. Hal ini dinilai cukup buruk untuk pengujian dataset selanjutnya karena akan banyak object yang miss karena avg loss yang tinggi.

Dari pengujian tersebut dapat disimpulkan hasil dari perbandingan gambar tersebut menunjukkan mAP (*mean Average Precision*) YOLOv3 61.5% dan YOLOv4 61.3% lalu untuk nilai Loss pada YOLOv3 0.35 dan YOLOv4 1.98 ini menunjukkan bahwa YOLOv3 lebih baik daripada YOLOv4 dikarenakan YOLOv3 cenderung baik jika object dan dataset yang cukup sedikit untuk melatih sedangkan YOLOv4 adalah hasil penyempurnaan dari YOLOv3 maka akan lebih baik jika dataset dan object lebih banyak daripada YOLOv3. Oleh karena itu pada pengujian ini kami memakai YOLOv3 untuk pengaplikasian di android.

4.2 Pengujian Skenario Kedua

Pengujian ini dilakukan untuk mengetahui kemampuan object detection pada YOLOv3 jika digunakan pada android. Pengujian menggunakan foto dari object yang sudah dilatih, ini dilakukan untuk mengetahui apakah ada perbedaan jika dideteksi di PC dan di android.



Gambar 4. 3 Object detection pada PC & Android

Terlihat pada Gambar 4.3 model dapat mengenali object yang sudah dilatih dan menampilkan nama dari object tersebut, dan melihat perbedaan pada PC dan Android tidak terjadi perbedaan pendeteksi sama sama mengenali object tersebut sebagai pisang.

Pada Skenario ini dilakukan perbandingan antara akurasi terhadap pengujian di android dan diperangkat computer. Pengujian ini dilakukan sebanyak 30 kali untuk memastikan system dapat melakukan pengenalan object dengan benar. Berikut merupakan hasil dari pengujian object pada model akan ditunjukkan pada tabel di bawah ini:

Tabel 4. 1 Akurasi Perbandingan pada Android dan PC

Perangkat	Citra	Percobaan Berhasil	Akurasi (%)
Android	30	27	88%
PC	30	27	88%

Terlihat pada Tabel 4.1 Akurasi pada android dan PC tidak memiliki perbedaan ini dikarenakan file yang dipakai di android dipanggil tidak disimpan pada android. Maka oleh karena itu tidak ada perbedaan antara akurasi pada android dan PC. Pada pengujian ini dapat disimpulkan bahwa model YOLOv3 tidak berubah pada saat diuji pada PC dan Android karena model tidak dirubah hanya dipanggil melalui Application program interface (API) menggunakan ip address karena tidak ada perubahan model maka hasilnya akan tetap sama, beda halan jika model dimasukkan kedalam android tanpa menggunakan API maka model akan di compress lebih kecil dari model sebelumnya yang akan mempengaruhi model tersebut.

4.3 Pengujian Skenario Ketiga

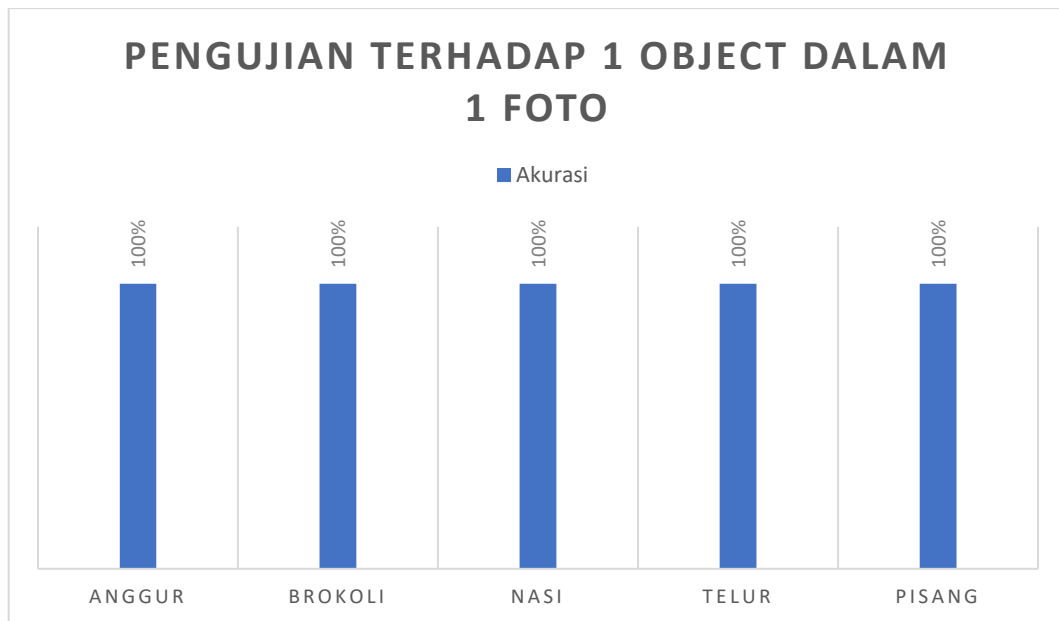
Pada scenario ini dilakukan pengujian akurasi deteksi pada citra yang terdiri dari pengujian 1, 3 dan 5 kelas dalam 1 image, untuk mengetahui nilai akurasi model, model diuji apakah bisa mendeteksi lebih dari satu object dalam satu image. Pengujian ini akan dilakukan dengan data uji yang berbeda beda dari 1 object, 3 object dan 5 object dalam satu citra. Jarak dan kondisi dataset dilakukan tidak melebihi jarak pandang kamera atau tidak melebihi 30cm. pengujian ini menggunakan object yang sesuai dengan object yang dilatih dilakukan untuk mengukur performansi dalam kondisi banyak object dalam satu image. Dipengujian ini nilai akurasi dihitung menggunakan persamaan sebagai berikut:

$$\frac{\text{Percobaan Berhasil}}{\text{Jumlah Percobaan}} \times 100\% = \text{Hasil Akurasi}$$

Berikut hasil pengujian skenario ini akan ditunjukkan pada tabel di bawah ini:

Tabel 4. 2 Pengujian dalam satu object satu foto

Nama Object	Jumlah object dalam image	Jumlah percobaan	Akurasi (%)
Anggur	1	30	100%
Brokoli	1	30	100%
Nasi	1	30	100%
Telur	1	30	100%
Pisang	1	30	100%



Gambar 4. 4 Grafik pengujian satu object satu image terhadap akurasi system

Dari **Gambar 4.4** dan **Tabel 4.3** diperlihatkan hasil akurasi system dari pengujian dalam satu object pada satu foto. Dalam pengujian ini menghasilkan akurasi yang baik, sebanyak 30 kali percobaan untuk setiap object nya. Dalam percobaan 1 kelas, object mendeteksi keberhasilan 100%.

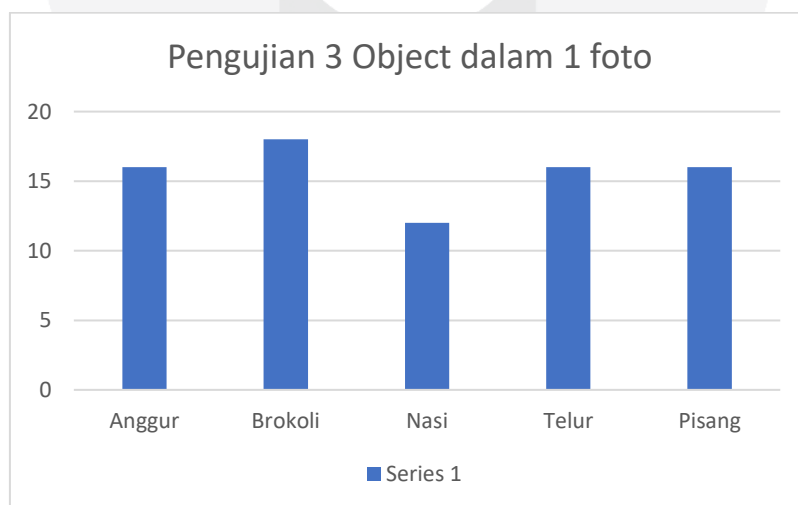
$$\frac{30 \text{ berhasil}}{30 \text{ percobaan}} \times 100\% = 100\%$$

Sehingga dapat disimpulkan hasil dari pengujian satu object dalam satu foto menghasilkan akurasi yang maksimal yaitu 100% dalam 30 kali percobaan artinya performa system sangat baik pada saat pengujian satu object satu foto.

Selanjut nya pada pengujian ini dilakukan menggunakan 3 Object dalam 1 citra berikut hasil percobaannya:

Tabel 4. 3 Percobaan 3 Kelas dalm 1 foto Jumlah terdeteksi

Nama Object	Jumlah terdeteksi	Jumlah Tidak Terdeteksi
Anggur	16	2
Brokoli	18	0
Nasi	12	6
Telur	16	2
Pisang	16	2



Gambar 4. 5 Grafik Percobaan 3 kelas dalam 1 foto

Dari **Gambar 4.5** dan **Tabel 4.4** bisa dilihat bahwa hasil pengujian banyak tidak terdeteksi pada Nasi hanya 12 dari 18 pendeteksi yang dalam 18 percobaan selalu melibatkan nasi ini. Nilai akurasi pada percobaan tiga kelas

dalam satu foto adalah dengan menjumlahkan jumlah yang terdeteksi dengan jumlah semua yang harus dideteksi maka hasilnya sebagai berikut:

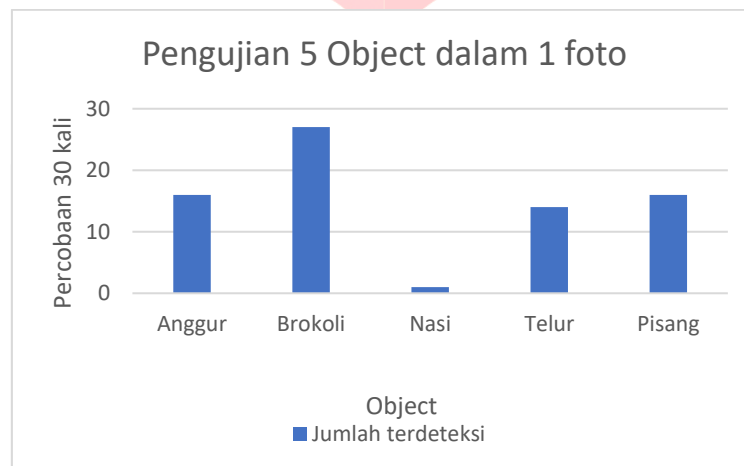
$$\frac{79 \text{ Jumlah Terdeteksi}}{90 \text{ Jumlah semua terdeteksi}} \times 100\% = 89\%$$

Maka Analisa dari Percobaan ini adalah menurunnya performa akurasi pada saat object digabungkan dalam satu foto, ini dikarenakan dalam dataset tidak ada data yang dalam kondisi dari object tersebut dalam satu kondisi yang sama. Maka mesin akan kesulitan dalam memahami kondisi tersebut dan mengapa nasi buruk dalam mendeteksi dalam percobaan ini? Karena nasi mempunyai kondisi object yang tidak jelas dalam bentuknya karena model ini adalah object detection maka system akan kebingungan dalam mendeteksi nasi.

Setelah mendeteksi satu object dalam satu foto dan 3 object dalam satu foto maka ini pengujian terakhir terhadap pengujian 5 object dalam 1 foto, berikut hasil percobaannya:

Tabel 4. 4 Percobaan 5 Object dalam 1 foto

Nama Object	Jumlah Terdeteksi	Jumlah tidak terdeteksi	Jumlah Percobaan
Anggur	16	14	30
Brokoli	27	3	30
Nasi	1	29	30
Telur	14	16	30
Pisang	16	14	30



Gambar 4. 6 Grafik Jumlah Terdeteksi Pengujian 5 Object dalam 1 foto

Pada **Gambar 4.6** dan **Tabel 4.5** terlihat bahwa pengujian pada tahap ini adalah banyak sekali yang terdeteksi dan apalagi pada nasi yang sangat jelek pada pendeteksi pada saat pengujian dan akurasi. Pada Tabel diatas menunjukkan keberhasilan pendeteksi mengalami penurunan akurasi mencapai 68%. Ini dihitung berdasarkan perhitungan yang sama dengan sebelumnya yaitu:

$$\frac{102 \text{ Percobaan Berhasil Terdeteksi}}{150 \text{ Jumlah Percobaan}} \times 100\% = 68\%$$

Maka dari hasil pengujian scenario ketiga didapatkan analisis pada percobaan satu object dari 1 foto adalah 100%, 3 object dalam 1 foto adalah 88% dan 5 object dalam 1 foto 68%. Terlihat bahwa semakin banyak object dalam 1 foto, performa semakin menurun ini dikarenakan kondisi dataset tidak ada dalam kondisi Bersama semua dataset dilabeling pada satu object terhadap satu foto dan semakin banyaknya object dalam satu foto membuat jarak semakin jauh dan object semakin mengecil yang membuat performa berkurang. Dan pada 5 object dalam 1 foto. Terlihat bahwa nasi sangat buruk pendeteksiannya ini dikarenakan bentuk nasi yang tidak bisa diprediksi. Pada kondisi dataset nasi tidak dibuat nasi harus berbentuk. Jadi dikarenakan ini object detection system susah untuk mendeteksi nasi. Hal lain yang mempengaruhi performa menurun adalah dataset seperti telur dan nasi adalah dataset sendiri yang tidak disediakan oleh OIDv4 dataset. hal ini mungkin terjadi pada tidak tepatnya penlabelan image, dan dataset pada pelatihan kurang banyak seharusnya bisa mencapai 1000 image. Sedangkan pada pengujian ini hanya 500 image.

5. Kesimpulan dan Saran

Setelah dilakukan pengujian dan analisis terhadap Aplikasi yang dibuat dapat disimpulkan beberapa hal berikut:

1. Pada penelitian ini berhasil membuat Aplikasi perhitungan nutrisi menggunakan Aplikasi Android. Aplikasi akan mendeteksi makanan yang difoto dengan sesuai dan menampilkan kandungan nutrisi

dari makanan yang berhasil di deteksi oleh file model pelatihan. Namun makanan yang difoto mengalami kesalahan mendeteksi akan tetap menampilkan kadar nutrisi tergantung mesin mengenal makanan itu apa, beda halnya jika makanan itu tidak terdeteksi maka makanan tersebut tidak akan menampilkan kadar nutrisi makanan tersebut.

2. Sistem model YOLOv3 akan terhubung dengan aplikasi android melalui API yang dibuat. API digunakan untuk menjalankan file pelatihan di android dikarenakan tidak memungkinkan menyimpan file pelatihan yang besar didalam aplikasi android dan ini membuktikan bahwa model YOLOv3 bisa digunakan untuk aplikasi android
3. Hasil pengujian Akurasi yang dilakukan menunjukan berupa kondisi pendeteksi tidak berjalan dengan baik dikarenakan banyaknya kelas didalam 1 gambar. Kondisi ini disebabkan oleh gambar yang dilatih tidak ada kondisi dimana makanan yang dideteksi dalam satu gambar yang bersamaan dan kurangnya dataset.

Saran

Sistem perhitungan nutrisi pada penelitian tugas akhir ini masih dapat dikembangkan dengan tujuan meningkatkan performansi kinerja system akurasi pendeteksi makanan dan perhitungan dengan berat asli makanan tersebut. Oleh karena itu terdapat beberapa saran dari penulis untuk pengembangan system selanjutnya:

1. Pengambilan dataset diharapkan lebih maksimal dengan hampir 1000 dataset dalam satu jenis makanan dan penlabelan gambar lebih teliti.
2. Data yang diambil seragam dan lebih spesifik seperti jenis memasak telur, jenis pisang dikarenakan berbeda jenis berbeda kadar nutrisinya.
3. Hasil perhitungan nutrisi akan dipengaruhi oleh berat makanan oleh karena itu maka lebih baik dikonfigurasi dengan timbangan.
4. System diharapkan bisa digunakan dengan custom dataset makanan lebih baik dan variasi makanan lebih banyak selain objek pada tugas akhir ini.

Referensi

- [1] K. Kusmiyati, "Nutrisi Di Awal Perkembangan," *J. Pijar Mipa*, vol. 7, no. 1, 2012, doi: 10.29303/jpm.v7i1.86.
- [2] A. . Fallis, "Kebutuhan Nutrisi," *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2013, doi: 10.1017/CBO9781107415324.004.
- [3] A. Penghitungan, K. Gizi, L. Berbasis, S. Android, R. A. Supono, and Y. D. Wulandari, "(No Title)."
- [4] N. Publikasi, I. Eka, A. Novita, and H. Sulistyanto, "PENGEMBANGAN APLIKASI UNTUK MENGETAHUI KEBUTUHAN JUMLAH KALORI," 2015.
- [5] L. D. Asih and M. Widyastiti, "Meminimumkan Jumlah Kalori di Dalam Tubuh dengan Memperhitungkan Asupan Makanan dan Aktivitas Menggunakan Linear Programming," *Ekologia*, vol. 16, no. 1, pp. 38–44, 2016.
- [6] A. D. Fitriyanti, "APLIKASI PENGHITUNG KALORI TERBAKAR SAAT BEROLAHRAGA SEPEDA MENGGUNAKAN GLOBAL POSITIONING SYSTEM (GPS) BERBASIS ANDROID."
- [7] H. Fauzi, N. A. Darsono, and . B. H., "Analisis Kalkulasi Body Mass Index Dengan Pengolahan Citra Digital Berbasis Aplikasi Android," *J. Elektro dan Telekomun. Terap.*, vol. 5, no. 2, p. 693, 2019, doi: 10.25124/jett.v5i2.1395.
- [8] L. Verdiana and L. Muniroh, "Kebiasaan Sarapan Berhubungan Dengan Konsentrasi Belajar Pada Siswa Sdn Sukoharjo I Malang," *Media Gizi Indones.*, vol. 12, no. 1, p. 14, 2018, doi: 10.20473/mgi.v12i1.14-20.
- [9] M. Yunus, "Perbandingan Metode-metode Edge Detection untuk Proses Segmentasi Citra Digital," *J. Teknol. Inf.*, vol. 3, no. 2, pp. 146–160, 2012.
- [10] "Android Developers." [Online]. Available: <http://developer.android.com/index.html>.
- [11] D. Manajang, S. Dompie, and A. Jacobus, "Implementasi Framework Tensorflow Object Detection Dalam Mengklasifikasi Jenis Kendaraan Bermotor," *J. Tek. Inform.*, vol. 15, no. 3, pp. 171–178, 2020.
- [12] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," 2018, [Online]. Available: <http://arxiv.org/abs/1804.02767>.
- [13] B. Adi Pranata, A. Hijriani, and A. Junaidi, "Perancangan Application Programming Interface (Api) Berbasis Web Menggunakan Gaya Arsitektur Representational State Transfer (Rest) Untuk Pengembangan Sistem Informasi Administrasi Pasien Klinik Perawatan Kulit," *J. Komputasi*, vol. 6, no. 1, pp. 33–42, 2018, doi: 10.23960/komputasi.v6i1.1554.