

Penerapan *ESP32-CAM* dan *TinyML* dalam Klasifikasi Gambar Buah dan Sayuran

Johni Revormasi Ziliwu^{1*}, Gogor. C. Setyawan², Haeni Budiati³
 Informatika, Universitas Kristen Immanuel, Yogyakarta, Indonesia
 *e-mail *Corresponding Author*: jrevormasi@gmail.com

Abstrak

The classification of fruit and vegetable images still requires high costs, posing challenges in developing efficient solutions. Fruits and vegetables are crucial in healthy food, so efficiency in managing their images impacts the agricultural industry. The aim of this study is to apply ESP32-CAM and TinyML in Fruit and Vegetable Image Classification with efficiency and low cost. The method involves Edge Impulse Studio and training the model with Convolutional Neural Network (CNN). From the testing, the F1 score accuracy reached 68.3% for each class of fruit and vegetables. From the demonstration using ESP32-CAM, the obtained accuracies are Apple (89%), Banana (91%), Orange (89%), Carrot (83%), and Cabbage (66%). The results indicate that applying ESP32-CAM and TinyML has the potential to improve efficiency and reduce costs in managing images.

Keywords: *ESP32-CAM; TinyML; Image Clasification; Fruit and Vegetables; Solution Efficiency;*

Abstrak

Pengklasifikasian gambar buah dan sayuran masih memerlukan biaya yang tinggi, sehingga menghadirkan kesulitan dalam mengembangkan solusi yang efisien. Buah dan sayuran penting dalam makanan sehat, sehingga efisiensi dalam pengelolaan gambar mereka berdampak pada industri pertanian. Tujuan penelitian ini adalah menerapkan ESP32-CAM dan TinyML dalam Klasifikasi Gambar Buah dan Sayuran dengan efisiensi dan biaya rendah. Metode melibatkan *Edge Impulse Studio* dan pelatihan model dengan *Convolutional Neural Network (CNN)*. Dari pengujian, akurasi F1 score mencapai 68.3% untuk setiap kelas buah dan sayuran. Dari demonstrasi menggunakan ESP32-CAM, akurasi yang diperoleh adalah Apel (89%), Pisang (91%), Jeruk (89%), Wortel (83%), dan Kubis (66%). Hasil menunjukkan penerapan ESP32-CAM dan TinyML memiliki potensi untuk meningkatkan efisiensi dan mengurangi biaya pengelolaan gambar.

Kata kunci: *ESP32-CAM; TinyML; Klasifikasi Gambar; Buah dan Sayuran; Efisiensi Solusi*

1. Pendahuluan

Di era teknologi saat ini yang semakin canggih, pengembangan sistem sangat mudah untuk dilakukan, termasuk dalam pengembangan sistem pengenalan gambar telah menjadi salah satu terobosan penting dalam berbagai bidang, termasuk pengklasifikasian gambar buah dan sayuran[1]. Buah dan sayuran merupakan salah satu komponen utama dalam makanan sehat dan bergizi yang mengandung kaya akan nutrisi seperti vitamin C, vitamin A, mineral, kalsium, dan zat besi [2].

Dalam pengelolaan buah dan sayuran saat ini, termasuk dalam pengklasifikasi gambar buah dan sayuran masih membutuhkan biaya yang tinggi terutama dalam pengadaan perangkat keras, pengembangan model, dan pemrosesan data, yang sudah barang tentu membutuhkan sebuah teknologi dalam pengklasifikasian gambar buah dan sayuran yang berbiaya rendah, efektif dan efisien [3].

Teknologi ESP32-CAM dan TinyML memberikan kontribusi dalam mengatasi masalah ini untuk meningkatkan efisiensi dalam pengklasifikasian gambar buah dan sayuran. ESP32-CAM adalah mikrokontroler yang dilengkapi dengan kamera dan modul WIFI, memungkinkan untuk pengambilan gambar secara langsung dan akurat[4]. ESP32-CAM teknologi yang sangat relatif lebih murah dibandingkan dengan teknologi lainnya. TinyML adalah sebuah teknologi machine learning perangkat daya rendah dan sumber daya terbatas, sehingga dapat

melakukan pengenalan gambar secara lokal tanpa harus mengirim data keserver [5]. Penelitian ini bertujuan untuk menggambarkan bagaimana penerapan ESP32-CAM dan TinyML dalam klasifikasi gambar buah dan sayuran serta menunjukan hasil dari proses klasifikasi gambar buah dan sayuran.

Di dalam industri pertanian dan makanan juga tentunya sangat membutuhkan teknologi dalam klasifikasi gambar buah dan sayuran [6] untuk pengelolaan stok [7], identifikasi penyakit, dan penentuan kematangan. Namun, proses klasifikasi masih membutuhkan biaya yang besar [8]. Dengan adanya teknologi ESP32-CAM dan TinyML, dapat digunakan dalam proses klasifikasi gambar buah dan sayuran di bidang pertanian dan makan, menjadi lebih hemat biaya, mudah dan efisien.

2. Tinjauan Pustaka

Menurut V. Gutti dan R. Karthi pada penelitiannya "*Real Time Classification of Fruits and Vegetables on Low Power Embedded Devices Using TinyML*" [9]. Pada penelitiannya pengklasifikasian pada buah dan sayuran menggunakan model *MobileNetV1* dan *ESP32-CAM* dengan kumpulan data gambar 360. Dari kedua model yang digunakan, hasil akhir klasifikasi buah dan sayuran pada model *MobileNetV1* mencapai akurasi pengujian 94% dari 17 kelas data uji dan hasil akhir dengan model *ESP32-CAM* adalah 77% dengan kecepatan inferensi 51 ms dan penggunaan memori 66,1 kb. Hasil percobaan pada penelitian ini menunjukan bahwa mikrokontroler *ESP32-CAM* berbiaya rendah dapat digunakan untuk klasifikasi produk pertanian.

Menurut Hanung Pangestu Rahman, Rahmat dan Jamaludin Indra pada penelitiannya "Penerapan *Convolution Neural Network* pada Timbangan Pintar Menggunakan *ESP32-CAM*" [10]. Pada penelitiannya ini membahas tentang penggunaa *ESP32-CAM* yang relatif lebih mura dari pada teknologi *Raspberry Pi* pada timbangan pintar untuk mendapatkan nilai berat pada suatu benda menggunakan metode *Convolutional Neural Network (CNN)* dan *loadcell* yang dilengkapi dengan modul *HX711*. Dalam penelitian ini dataset dikumpulkan 600 data, jenis sayuran 150 data dan pada kelas data training terdiri dari tomat, kubis, wortel, dan kentang. Dari penelitian ini setelah dilakukan pengujian, hasil akhir akurasi klasifikasi dari penggunaan *ESP32-CAM* pada timbangan pintar dari data sebesar 90% dan selisih rata-rata alat dibangun sebesar 0.8 gram dibandingkan timbangan digital merek *SF-400*.

Penelitian yang dilakukan Yanri Bili Eliezer, Naikson F Saragih, Arina Prima Silalahi, Surianto Sitepu, Asaziduhu Gea, pada penitianya "Perancangan Alat Pendekteksi Kematangan Buah Nanas Dengan Menggunakan Mikrokontroler Dengan Metode *Convolutional Neural Network (CNN)*" [11]. Pada penelitian ini membahas tentang mendekteksi kulit buah nanas menggunakan *ESP32-CAM* sebagai teknologi untuk mengambil gambar objek buah nanas dan menggunakan metode *CNN*. Hasil akhir dari penelitian ini pendeksti kematangan buah nanas menggunakan mikrokontroler metode *CNN* yaitu hasil akhir nilai akurasi tertinggi 86%, akurasi terendah 80% dan total semua rata-rata akurasi 83,33%.

Penelitan yang dilakukan Associative professor, Arulmurugan L Raghav, Nikhil V Kumar P, Bharath S, penelitian "*Animal Intrusion Detection Using ESP32 Cam and Open CV*" [12]. Pada penelitian ini membahas tentang penggunaan *ESP32 Cam* dan *Open CV* dalam mendeteksi intrusi hewan dalam jarak jauh. Penelitian ini menggunakan algoritma terlatih berbasis *yolov5* untuk deteksi objek. Hasil akhir mencapai akurasi maksimum algoritma dilatih dengan ribuan foto binatang berganda kali hingga akurasi yang diinginkan (~98%).

Penelitian yang dilakukan Hendrick, Hanifa Fitri, dan Ivan Finiel Hotmartua Bagaring pada penelitian "Pemanfaatan *ESP32-CAM* Untuk Mengukur Ketinggian Air Menggunakan Metode *Image Processing*" [13]. Penelitian ini membahas pemanfaatan *ESP32-CAM* untuk mengukur ketinggian air menggunakan metode *image processing*. Dalam pemrosesan gambar, hasil didapatkan dari perubahan pixel, yaitu diakibatkan karena naik turunnya objek perubahan tinggi air sungai dan menggunakan rumus regresi polinomial. Hasil akhir dari pemanfaatan *ESP32-CAM* mengukur ketinggian ari menggunakan metode *image processing* ada 2 hasil akhir yaitu pengukuran akurat dan pengukuran tidak akurat. Pada ketinggian 1 cm sampai 7 cm nilai pengukuran keakuratan kesalahan relatif sebesar 1,26% dan pada ketinggian 8 cm nilai pengukuran tidak akurat kesalahan relatif sebesar 250,31 %.

Berdasarkan beberapa penelitian terdahulu yang dilakukan dalam penelitian penerapan atau penggunaan *ESP32-CAM* dalam klasifikasi pada gambar memberikan salah satu solusi yang baik dalam melakukan pengklasifikasian pada gambar dengan efektif dan efisien. Pada

penelitian-penelitian sebelumnya penerapan atau penggunaan ESP32-CAM lebih berfokus pada deteksi objek, pengukuran, dan klasifikasi lainnya.

Pada penelitian ini, penerapan ESP32-CAM dan TinyML Dalam klasifikasi Gambar Buah dan Sayuran lebih berfokus pada pengklasifikasian gambar buah dan sayuran yang dimana menggunakan metode pelatihan model yaitu *Edge Impulse Studio* sebagai pendekatan yang lebih canggih dan spesifik untuk tugas klasifikasi gambar buah dan sayuran. Selain itu, pada penelitian ini juga menggunakan metode *Convolutional Neural Network* (CNN) sebagai metode klasifikasi pada gambar untuk mengukur akurasi pada gambar buah dan sayuran dan untuk menunjukan keefektian dalam pengenalan pola pada gambar buah dan sayuran. Dengan integrasi teknologi ini menghasilkan solusi yang lebih canggih dan efisien dalam penerapan ESP32-CAM dan TinyML dalam klasifikasi buah dan sayuran, menjadi sebagai perubahan yang signifikan dalam pengembangan sistem cerdas berbiaya rendah dan efektif.

3. Metodologi Penelitian

Penelitian ini dilakukan dengan mengikuti beberapa tahapan atau prosedur yaitu dimulai dari: Studi Literatur dan Analisis Kebutuhan, Desain *Hardware* dan *Software*, Pengumpulan Dataset dan Model Training, dan terakhir pengujian akurasi dan validasi dari hasil Model Training.



Gambar 1 . Tahapan Metode Penelitian

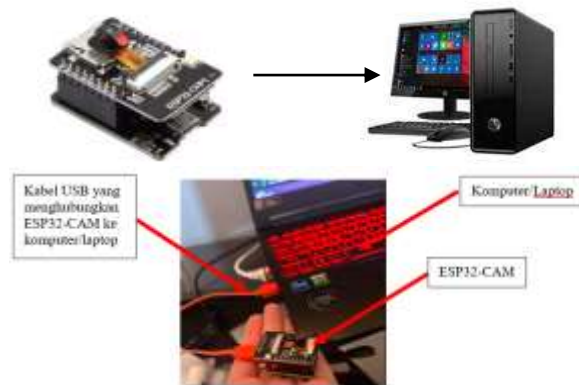
3.1 Studi Literatur dan Analisis Kebutuhan

Tahapan ini merupakan langkah awal dalam penelitian yaitu peneliti melakukan studi literatur atau riset pada penelitian terdahulu untuk mendapatkan informasi dan teori pendukung. Dengan mengumpulkan berbagai informasi dan teori pendukung dari penelitian terdahulu maka peneliti mampu melakukan analisis kebutuhan untuk penelitian. Berikut adalah hal-hal yang menjadi kebutuhan dalam penelitian ini:

- 1) *Platform* pengembangan AI atau Kecerdasan Buatan. Dalam hal ini platform yang digunakan adalah Edge Impulse Studio yang tersedia secara online di edgeimpulse.com
- 2) *Microcontroller* ESP32-CAM sebagai perangkat keras untuk melakukan pengambilan gambar buah dan sayur yang akan diklasifikasi.
- 3) Arduino IDE sebagai *software* pengembangan *microcontroller* untuk memprogram ESP32-CAM.
- 4) Dataset gambar buah dan sayur. Dataset dapat digunakan secara gratis di situs www.kaggle.com yang disediakan oleh komunitas[14].
- 5) Sebuah komputer dengan spesifikasi RAM 4GB dan sistem operasi Windows 10.

3.2 Desain *Hardware* dan *Software*

Sesuai dengan analisis kebutuhan, maka hardware yang dibutuhkan adalah sebuah ESP32-CAM dan sebuah komputer. Desainnya cukup dengan menghubungkan komputer dan ESP32-CAM dengan sebuah kabel USB yang dapat dilihat pada Gambar 2.

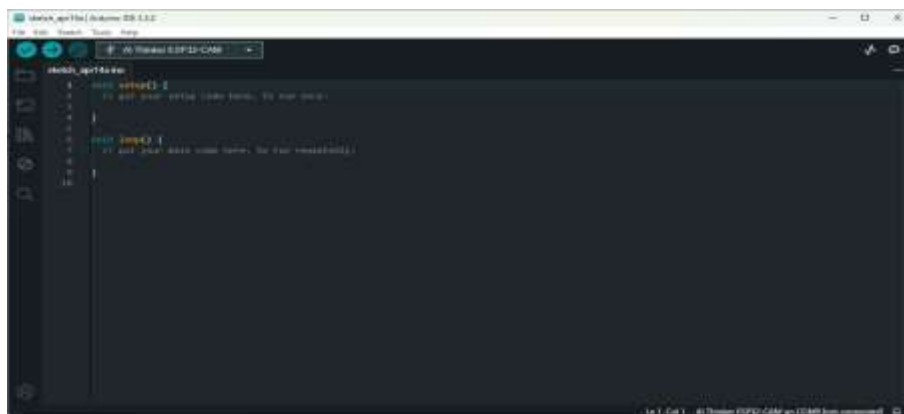


Gambar 2 Desain *Hardware* ESP32-CAM dan Komputer

Software yang digunakan adalah *Edge Impulse Studio* sebagai AI development platform dan *Arduino IDE* untuk memprogram ESP32-CAM. Berikut adalah tampilan dari *Edge Impulse Studio* dan *Arduino IDE*.



Gambar 3 Tampilan dashboard *Edge Impulse Studio*



Gambar 4 Tampilan *Arduino IDE*

3.3 Pengumpulan Data dan *Model Training*

1) Pengumpulan Data

Pada tahap ini data yang dikumpulkan adalah *image dataset* atau gambar yang akan dijadikan model training untuk klasifikasi objek. Dalam hal ini, gambar yang akan dikumpulkan

adalah gambar buah dan sayur. Kumpulan sampel gambar buah dan sayur dapat diambil di situs kaggle.com yang disediakan secara gratis oleh komunitas. Pada penelitian ini peneliti akan menggunakan 5 jenis objek yang akan diklasifikasi yaitu: apel, pisang, jeruk, kubis, dan wortel. Berikut adalah jumlah data sampel yang digunakan untuk training model:

Tabel 1 Jumlah dataset sampel untuk model *training* dan *testing*

Objek	Train	Test
Apel	695	196
Pisang	704	200
Kubis	539	149
Wortel	1,113	265
Jeruk	907	284
Total dataset:		1,709 sampel

Jumlah dataset sampel ditentukan otomatis oleh sistem *Edge Impulse Studio* saat memberi label gambar yang telah diunggah.

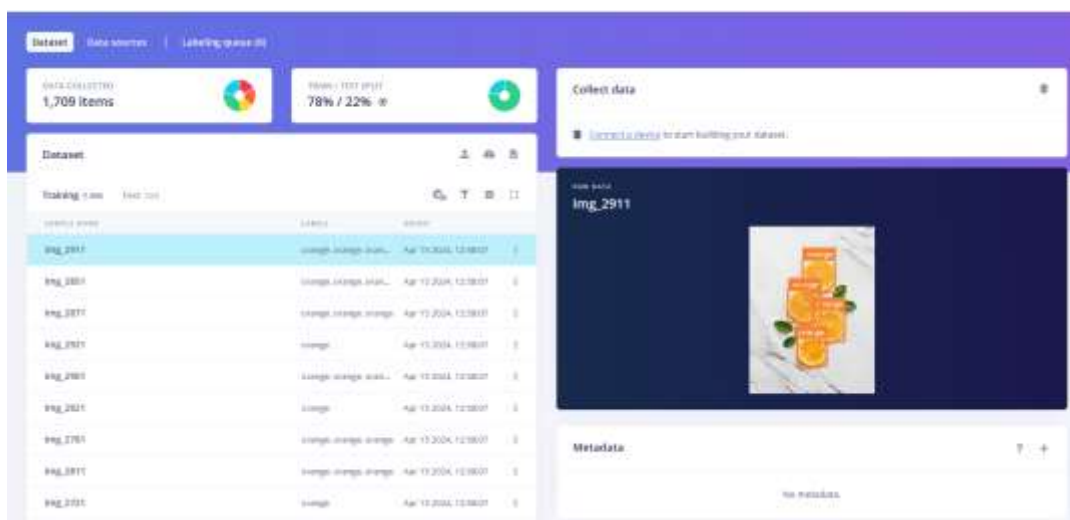


Gambar 5 Data Sampel Buah dan Sayur

2). Model Training

a. Persiapan Training Model pada *Edge Impulse Studio*

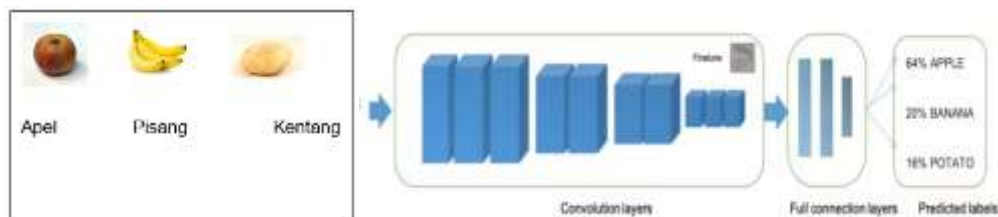
Setelah data sampel diunggah ke *Edge Impulse Studio*, langkah berikutnya adalah dengan melabeli gambar sampel satu per satu agar dapat dijadikan kelas model yang akan di-train. Setelah itu dilakukan pemerataan atau pembagian data antara sampel untuk *training* dan sampel untuk *testing*. Pemerataan dilakukan dengan membagi jumlah sampel training dan testing dengan rasio sekitar 80/20. Dalam penelitian kali ini, sistem otomatis membagi dataset sampel train dan test dengan rasio 78% dan 22%. Hal ini dilakukan untuk meningkatkan akurasi dari training model nanti.



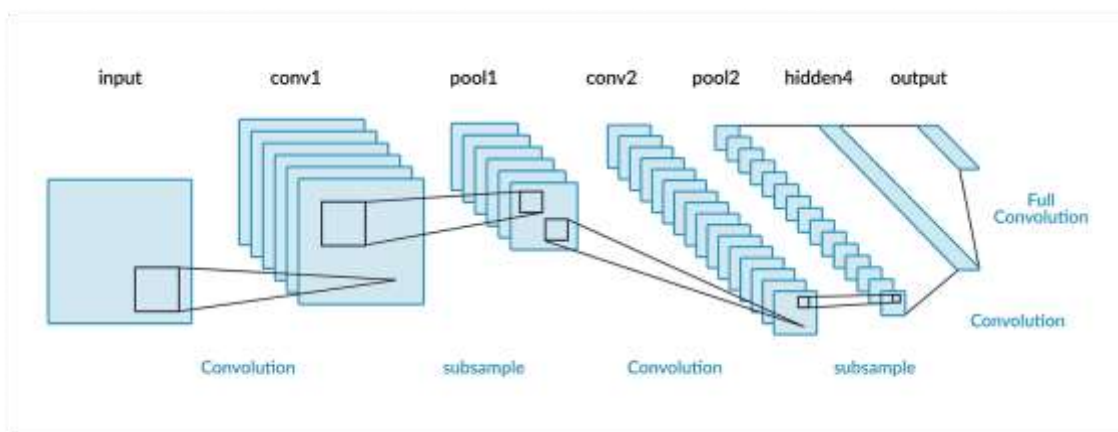
Gambar 6 Sampel data yang dikumpulkan dilabeli dan dibagi dengan rasio sekitar 80/20

b. Training model dengan menggunakan arsitektur *Convolutional Neural Network* (CNN)

Selanjutnya adalah melakukan model training. Algoritma yang digunakan adalah *Convolutional Neural Network* (CNN). *Convolutional Neural Network* (CNN) adalah metode yang menggunakan jaringan syaraf tiruan, yang dapat mengelola data citra[15]. Data gambar yang telah di dapatkan pada tahap pengumpulan data maka dilakukan *training dataset* menggunakan metode CNN dengan mengubah ukuran data gambar menjadi 96 x 96 dan melakukan data *augmentation* [16]. *Augmentation* adalah metode yang dapat membantumeningkatkan akurasi model pembelajaran mesin[17]. Berikut arsitektur *training dataset* pada penelitian ini dapat dilihat pada gambar 7.



Gambar 7. Arsitektur Training Dataset Metode CNN



Gambar 8. Tahapan Algoritma CNN

Pada gambar 8 Arsitektur Training Dataset Metode CNN, pertama data gambar yang sudah ada akan masuk pada tahap *Convolution layers*. Tahapan *Convolution layers* adalah melakukan pengenalan pada objek melalui filter yang digeser di atas input dan melakukan operasi perkalian matriks pada setiap lokasi, hasilnya akan masuk ke dalam peta fitur[18]. Setelah tahapan *Convolution layers* selanjutnya masuk pada tahap *Full connection layers*, tahap *Full connection* adalah tahap yang berfungsi untuk transformasi dimensi pada data gambar yang dari tahap *Convolution layers* untuk diklasifikasi secara linear[19]. Dengan melewati tahapan proses training dataset menggunakan metode CNN, maka data gambar yang telah menjadi model dapat digunakan pada proses klasifikasi menggunakan ESP32-CAM.

Pada penelitian ini, semua proses pelatihan model dengan menggunakan algoritma CNN akan dilakukan secara otomatis oleh sistem *Edge Impulse Studio* dengan menggunakan library Keras dan *TensorFlow*. Proses training model menggunakan algoritma CNN tidak perlu dilakukan manual oleh pengguna.

Neural network architecture

```

1 sys.path.append('./resources/libraries')
2 import os
3 import tensorflow as tf
4 from tensorflow.keras.optimizers.legacy import Adam
5 from tensorflow.keras.applications import MobileNetV2
6 from tensorflow.keras.layers import BatchNormalization, Conv2D, Softmax, Reshape
7 from tensorflow.keras.models import Model
8 from ei_tensorflow.constrained_object_detection import models, dataset, metrics, util
9 from ei_tensorflow.velo import train_keras_model_with_velo
10 from ei_shared.pretrained_weights import get_or_download_pretrained_weights
11 import ei_tensorflow.training
12
13 WEIGHTS_PREFIX = os.environ.get('WEIGHTS_PREFIX', os.getcwd())
14
15 def build_model(input_shape: tuple, weights: str, alpha: float,
16               num_classes: int) -> tf.keras.Model:
17     """ Construct a constrained object detection model.
18
19     Args:
20         input_shape: Passed to MobileNet construction.
21         weights: Weights for initialization of MobileNet where None implies
22                 random initialization.
23         alpha: MobileNet alpha value.
24         num_classes: Number of classes, i.e. final dimension size, in output.
25
26     Returns:
27         Uncompiled keras model.
28
29     Model takes (B, H, W, C) input and
30     returns (B, H//8, W//8, num_classes) logits.
31     """
32
33     #! First create full mobile_net_v2 from (Hw, Hw, C) input
34     #! to (Hw/8, Hw/8, C) output
35     mobile_net_v2 = MobileNetV2(input_shape=input_shape,

```

Gambar 9. Source code arsitektur CNN untuk training model

Pada *software Edge Impulse Studio*, tersedia beberapa arsitektur model yang dapat digunakan untuk deteksi objek. Salah satunya adalah model FOMO (*Faster Object More Object*) yang menggunakan metode CNN untuk melakukan deteksi objek. Model arsitektur ini memungkinkan AI melakukan *training model* dengan lebih cepat dari pada *MobileNet SSD* dengan hanya menggunakan RAM kurang dari 200 kb saja[20]. Parameter yang dipakai dalam model training adalah sebagai berikut:

- 1) Number of training cycles: 40
- 2) Learning rate: 0,001
- 3) Data augmentation: Yes
- 4) Validation set size: 20%
- 5) Batch size: 16

Setelah itu model sudah siap untuk di-training dan hasil akan ditampilkan setelahnya.

3.4 Pengujian Akurasi dan Validasi

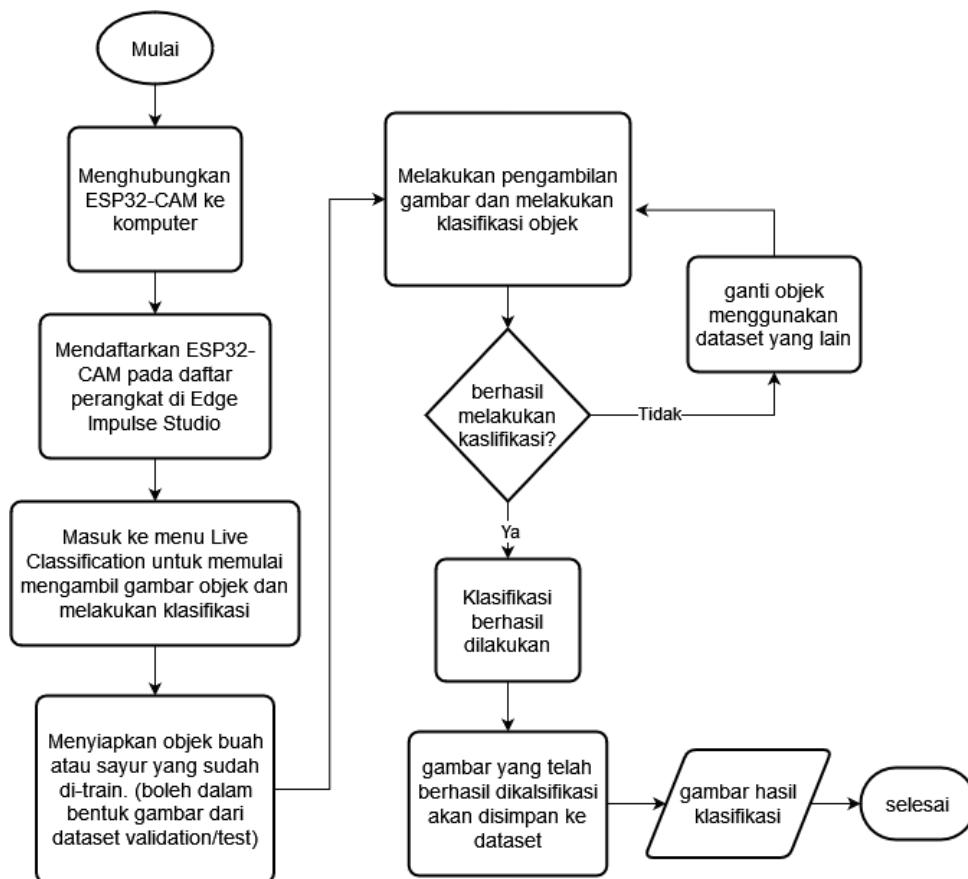
Saat proses training model, *Edge Impulse Studio* akan melakukan perhitungan akurasi dengan menghitung setiap hasil training model menggunakan *Confusion Matrix*. Matriks ini akan memvalidasi setiap objek yang diklasifikasi dengan objek lainnya serta latar belakang objek. Semakin rendah error maka semakin tinggi juga akurasi begitupun sebaliknya. Nilai akurasi setiap objek akan dirata-ratakan yang akan menjadi hasil akhir nilai akurasi yang disebut juga dengan *F1 score*. Jadi, semakin tinggi *F1 score* maka semakin akurat model yang sudah di-training. Rumus dari *F1 score* dapat dilihat pada persamaan berikut.

$$\begin{aligned}
 \text{F1 Score} &= \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}} \\
 &= \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \dots\dots\dots (1)
 \end{aligned}$$

Gambar 10 Rumus untuk menghitung F1 Score

Proses pengujian klasifikasi dilakukan dengan menggunakan *fitur Live Classification* yang disediakan oleh *Edge Impulse Studio*. Caranya adalah dengan mengikuti tahap berikut ini:

- 1) Menghubungkan ESP32-CAM ke komputer.
- 2) Mendaftarkan ESP32-CAM pada daftar perangkat di *Edge Impulse Studio*.
- 3) Masuk ke menu *Live Classification* untuk memulai mengambil gambar objek dan melakukan klasifikasi.
- 4) Menyiapkan objek buah/sayur yang sudah di-train. Objek dapat berupa gambar dari set validasi/test.
- 5) Melakukan pengambilan gambar dan melakukan klasifikasi objek.
- 6) Jika proses klasifikasi berhasil dilakukan maka gambar akan diberi label dan disimpan ke data set.
- 7) Jika proses klasifikasi gambar gagal, maka ganti objek dengan menggunakan dataset yang lain atau dengan memperbaiki kualitas pengambilan gambar.
- 8) Jika sudah mengganti objek ke dataset yang lain maka lakukan kembali langkah ke 5.
- 9) Proses klasifikasi berhasil dilakukan.



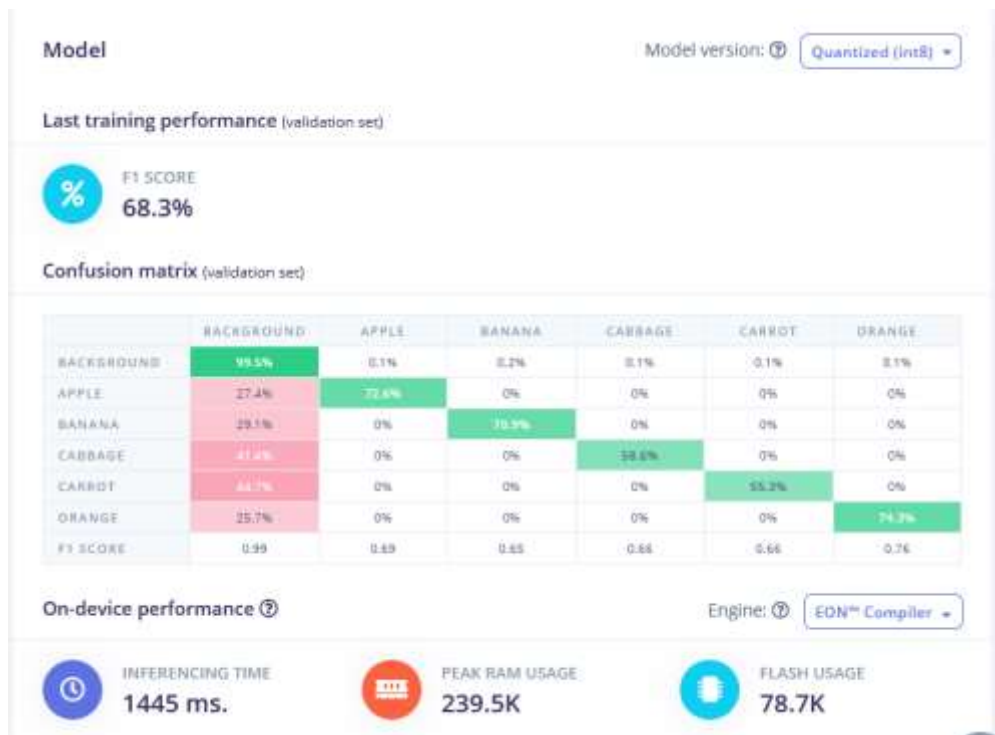
Gambar 11 Flowchart cara kerja klasifikasi objek menggunakan ESP32-CAM

4. Hasil dan Pembahasan

Berdasarkan hasil dari implementasi dari metode penelitian, maka dapat dibahas beberapa hasil dari penelitian yang telah dilakukan.

4.1 Hasil Model Training

Setelah proses model training selesai, Edge Impulse Studio dapat langsung memberikan hasil akurasi setiap objek yang di-train. Hasil training disajikan dalam bentuk *Confusion Matrix*. Berikut adalah hasil pengujian yang didapatkan peneliti saat training model selesai.



Gambar 12 Hasil Model Training dalam bentuk *Confusion Matrix*

Dari matriks tersebut hasil yang didapatkan adalah sebagai berikut:

- 1) Apel dengan akurasi 72.6%
- 2) Pisang dengan akurasi 70.9%
- 3) Kubis dengan akurasi 58.6%
- 4) Wortel dengan akurasi 55.3%
- 5) Jeruk dengan akurasi 74.3%
- 6) F1 Score 68.3%

Dari hasil training model di atas, dapat dibahas beberapa hal yaitu: akurasi buah apel, pisang dan jeruk memiliki hasil yang cukup baik. Namun, untuk kubis dan wortel masih di bawah rata-rata F1 score. Hal ini disebabkan karena data sampel yang diunggah kurang bersih saat dilabeli. Hal lain yang mempengaruhi adalah penggunaan RAM yang kecil akibat keterbatasan *hardware* yang dijadikan target *deployment*, yaitu ESP32-CAM yang mana hanya memiliki RAM sebesar 520kb sehingga sistem mengompres model yang di-train agar dapat di-deploy ke ESP32-CAM.

4.2 Demonstrasi *Live Classification* pada ESP32-Cam

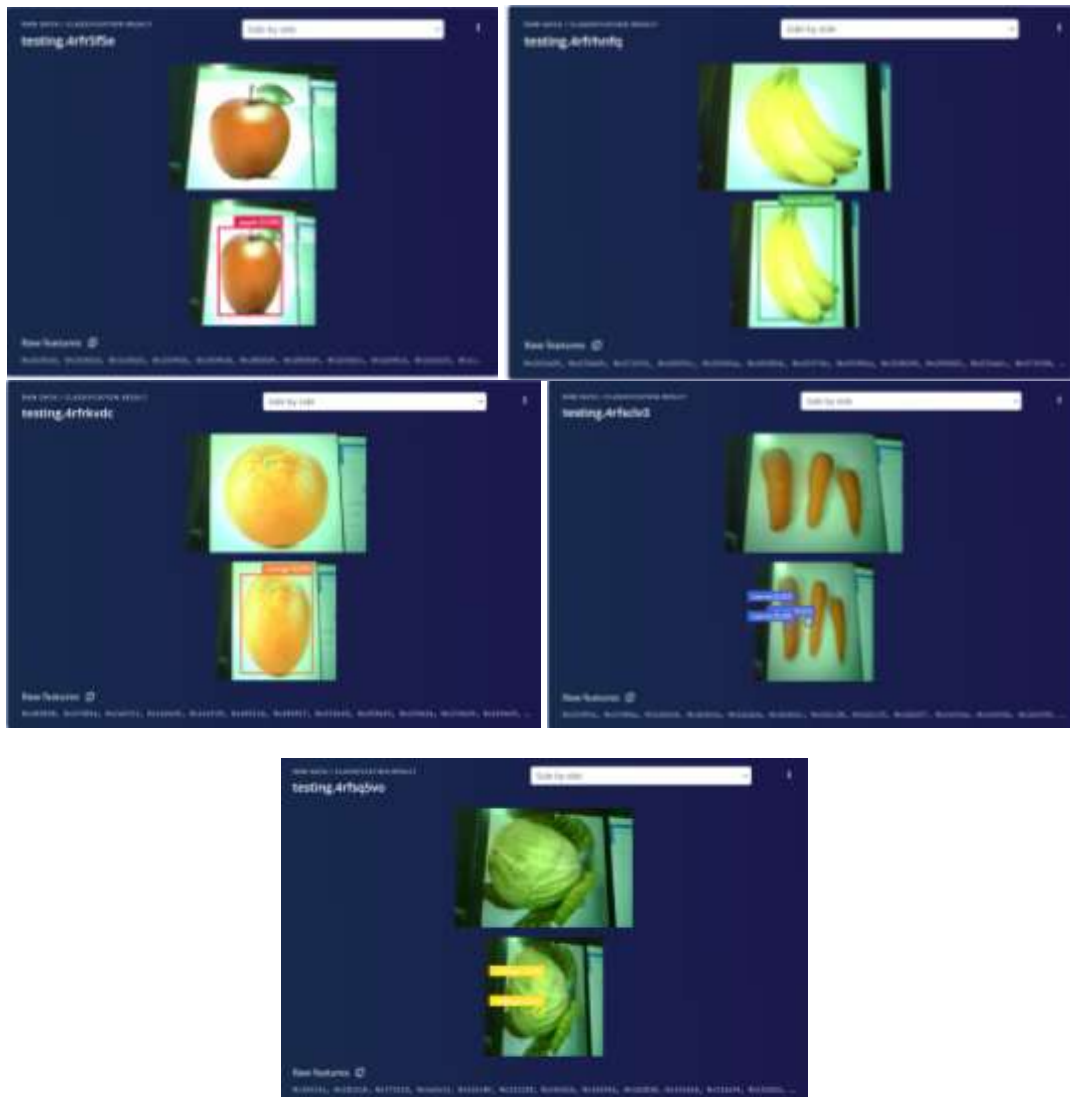
Model yang sudah selesai dapat di-deploy ke *device* atau perangkat yang akan digunakan untuk melakukan klasifikasi. Dalam hal ini perangkat yang akan digunakan adalah ESP32-CAM. Untuk men-deploy model maka perlu dibuat library Arduino yang akan ditulis ke dalam ESP32-CAM. Untuk men-deploy dapat dilakukan dengan masuk ke menu *Deployment* pada *Edge Impulse Studio* kemudian memilih perangkat yang akan digunakan. Setelah itu library Arduino akan di-generate oleh *Edge Impulse* yang jika sudah selesai dibuat dapat diunduh. Setelah itu library bisa dibuka pada Arduino IDE yang kemudian bisa diunggah ke ESP32-CAM. Setelah program berhasil di-deploy ke ESP32-CAM, maka langkah selanjutnya adalah melakukan percobaan demonstrasi klasifikasi objek dengan menggunakan ESP32-CAM pada gambar buah/sayur dari dataset testing seperti yang sudah dibahas pada bab 3 sebelumnya. Berikut adalah contoh demonstrasi pengambilan gambar buah/sayur menggunakan ESP32-CAM.

Pada pengujian ini, ESP32-CAM berhasil mendeteksi objek berupa gambar dataset testing yang ditampilkan di monitor komputer. Data hasil pengetesan dapat dilihat pada Tabel 2.

Tabel 2. Hasil Pengujian Sistem

Objek	Hasil Klasifikasi	Akurasi	Jumlah Percobaan
Apel	Apel (berhasil)	89%	1 kali (langsung terdeteksi)
Pisang	Pisang (berhasil)	91%	1 kali (langsung terdeteksi)
Jeruk	Jeruk (berhasil)	89%	1 kali (langsung terdeteksi)
Wortel	Wortel (cukup berhasil)	62%, 68%, 83%	3 kali (tidak langsung terdeteksi)
Kubis	Kubis (cukup berhasil)	66%, 61%	2 kali (tidak langsung terdeteksi)

Berikut adalah gambar hasil dari percobaan penangkapan gambar menggunakan ESP32-CAM beserta hasil klasifikasinya.



Gambar 14. Hasil percobaan klasifikasi objek apel, pisang, jeruk, wortel, dan kubis

Dari hasil percobaan demonstrasi pengambilan gambar objek menggunakan ESP32-Cam melalui *Edge Impulse Live Classification*, dapat disimpulkan bahwa akurasi hasil model training sangat mempengaruhi hasil pengujian klasifikasi pada praktik langsung. Jadi, semakin tinggi akurasi training model dalam hal ini ditunjukkan dalam bentuk *F1 score* dan *Confusion Matrix*, maka semakin bagus pula hasil klasifikasi objeknya.

```

00:07:09.419 -> Predictions (DSP: 9 ms., Classification: 794 ms., Anomaly: 0 ms.):
00:07:09.419 ->   apple (0.992188) [ x: 24, y: 32, width: 8, height: 16 ]
00:07:10.336 -> Predictions (DSP: 9 ms., Classification: 794 ms., Anomaly: 0 ms.):
00:07:10.380 ->   apple (0.964844) [ x: 24, y: 32, width: 8, height: 16 ]
00:07:11.302 -> Predictions (DSP: 9 ms., Classification: 793 ms., Anomaly: 0 ms.):
00:07:11.302 ->   apple (0.972656) [ x: 24, y: 32, width: 8, height: 16 ]
00:07:12.302 -> Predictions (DSP: 9 ms., Classification: 794 ms., Anomaly: 0 ms.):
00:07:12.302 ->   apple (0.960938) [ x: 24, y: 32, width: 8, height: 16 ]
00:07:13.246 -> Predictions (DSP: 9 ms., Classification: 793 ms., Anomaly: 0 ms.):
00:07:13.246 ->   apple (0.960938) [ x: 24, y: 32, width: 8, height: 16 ]
00:07:14.182 -> Predictions (DSP: 9 ms., Classification: 793 ms., Anomaly: 0 ms.):
00:07:14.224 ->   No objects found

```

Gambar 15. Tampilan hasil klasifikasi objek menggunakan Arduino IDE Serial Monitor

Berdasarkan pengujian dan demonstrasi di atas, maka dapat peneliti menyimpulkan beberapa poin yang membahas relevansi penelitian ini dengan penelitian terdahulu, yaitu:

- 1) Akurasi yang didapatkan pada penelitian ini tidak setinggi yang didapatkan penelitian terdahulu. Hal ini dimungkinkan karena terdapat perbedaan dataset sampel yang digunakan dan proses dalam pengolahan dataset sampel sebelum di-train.
- 2) Meskipun akurasi tidak terlalu tinggi, namun penelitian ini membuktikan bahwa ESP32-CAM dapat digunakan sebagai perangkat deteksi/klasifikasi objek meski memiliki spesifikasi yang cukup rendah yaitu hanya memiliki RAM sebesar 520 kb.
- 3) Arsitektur model CNN FOMO yang dimiliki Edge Impulse Studio sangat ringan yang terbukti dapat melakukan training model hingga dapat men-*deploy* model pada perangkat *microcontroller* dengan spesifikasi rendah.
- 4) *Platform AI Edge Impulse Studio* sangat memudahkan AI/ML engineer dalam membuat sebuah proyek AI dengan tersedianya *tools* dan *library* yang cukup lengkap, mudah dipakai, dan ringan.

5. Simpulan

Dari hasil penelitian ini penerapan ESP32-CAM dan TinyML dalam klasifikasi gambar dan sayuran dapat disimpulkan bahwa ESP32-CAM dan TinyML sebagai solusi yang efisien dan proses yang sangat mudah untuk klasifikasi gambar buah dan sayuran. ESP32-CAM merupakan perangkat yang sangat sederhana, murah, dan mudah untuk diprogram, memungkinkan pengguna untuk melakukan klasifikasi dengan biaya rendah dan tanpa memerlukan pengetahuan pemrograman yang mendalam. Dengan demikian, aplikasi ini memiliki potensi untuk dikembangkan dan diterapkan dalam berbagai bidang, seperti pertanian, pengolahan makanan dan lain-lain.

Daftar Referensi

- [1] A. Ciputra, D. R. I. M. Setiadi, E. H. Rachmawanto, and A. Susanto, "Klasifikasi Tingkat Kematangan Buah Apel Manalagi Dengan Algoritma Naive Bayes Dan Ekstraksi Fitur Citra Digital," *Simetris J. Tek. Mesin, Elektro dan Ilmu Komput.*, vol. 9, no. 1, pp. 465–472, Apr. 2018, doi: 10.24176/simet.v9i1.2000.
- [2] E. Priatiningsih, "Hubungan Pengetahuan Ibu Tentang Gizi Dengan Pengelolaan Makanan Sehat Keluarga Pada Anggota Lembaga Pemberdayaan Kesejahteraan Keluarga. Keluarga: Jurnal Ilmiah Pendidikan Kesejahteraan Keluarga, vol. 4, no. 1, pp. 101-108, 2018.
- [3] N. Ardila, "Aplikasi Pengelolaan Sayuran dan Buah-Buahan di Pizza Hut PTC Palembang. Other. Perpustakaan Universitas Katolik Musi Charitas, 2017.
- [4] N. Nikolov, & O. Nakov, "Research of secure communication of Esp32 IoT embedded system to. NET core cloud structure using MQTTS SSL/TLS. In 2019 IEEE XXVIII International Scientific Conference Electronics (ET), pp. 1-4, IEEE, 2019.
- [5] V. Tsoukas, E. Boumpa, G. Giannakas, & A. Kakarountas, "A review of machine learning and tinyml in healthcare. In Proceedings of the 25th Pan-Hellenic Conference on Informatics, pp. 69-73, 2021.
- [6] A. Ciputra, E.H. Rachmawanto, & A. Susanto, "Klasifikasi Tingkat Kematangan Buah Apel Manalagi Dengan Algoritma Naive Bayes Dan Ekstraksi Fitur Citra Digital". *Simetris: Jurnal Teknik Mesin, Elektro Dan Ilmu Komputer*, Vol. 9, no. 1, pp. 465-472, 2018.

- [7] D.T. Laswati, N.R.I. Sundari, & O. Anggraini, "Pemanfaatan kersen (*Muntingia calabura* L.) sebagai alternatif produk olahan pangan: sifat kimia dan sensoris". *JITIPARI (Jurnal Ilmiah Teknologi dan Industri Pangan UNISRI)*, vol. 2, no. 2, pp. 1-12, 2017
- [8] S.D.P. Bahari, & U. Latifa, "Klasifikasi Buah Segar Menggunakan Teknik Computer Vision Untuk Pendeteksian Kualitas Dan Kesegaran Buah. *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 7, no. 3, pp. 1567-1573, 2023.
- [9] V. Gutti and R. Karthi, "Real Time Classification of Fruits and Vegetables Deployed on Low Power Embedded Devices Using Tiny ML," 2022, pp. 347–359. doi: 10.1007/978-3-031-12413-6_27.
- [10] R. F. Hanung Pangestu Rahman, Jamaludin Indra, "Penerapan Convolutional Neural Network pada Timbangan Pintar Menggunakan ESP32-CAM," *J. Media Inform. Budidarma*, vol. 7, no. 2, pp. 1–9, 2023, doi: 10.30865/mib.v7i1.5469.
- [11] Y. B. E. Purba, N. F. Saragih, A. P. Silalahi, and ..., "Perancangan Alat Pendeteksi Kematangan Buah Nanas Dengan Menggunakan Mikrokontroler Dengan Metode Convolutional Neural Network (CNN)," *J. Ilm. Tek. ...*, vol. 2, no. 1, pp. 13–21, 2022, [Online]. Available: <https://ojs.fikom-methodist.net/index.php/methotika/article/view/43%0Ahttps://ojs.fikom-methodist.net/index.php/methotika/article/download/43/39>
- [12] A. L. Associative professor, N. V Raghav, and B. S. Kumar P, "Animal Intrusion Detection Using ESP32 Cam and Open CV," *Int. J. Innov. Sci. Res. Technol.*, vol. 8, no. 10, pp. 913–916, 2023, [Online]. Available: www.ijisrt.com913
- [13] H. Fitri and D. Ivan Finiel Hotmartua Bagariang, "Pemanfaatan Esp32-Cam Untuk Mengukur Ketinggian Air Menggunakan Metode Image Processing," *Semin. Nas. Terap. Ris. Inov. Ke-6 ISAS Publ. Ser. Eng. Sci.*, vol. 6, no. 1, pp. 762–769, 2020.
- [14] K. Seth, "Fruits and Vegetables Image Recognition Dataset," [kaggle.com](https://www.kaggle.com/datasets/kritikseth/fruit-and-vegetable-image-recognition). Accessed: Apr. 21, 2024. [Online]. Available: <https://www.kaggle.com/datasets/kritikseth/fruit-and-vegetable-image-recognition>
- [15] S. R. Suartika E. P, I Wayan, Wijaya Arya Yudhi, "Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) Pada Caltech 101," *J. Tek. ITS*, vol. 5, no. 1, pp. 76-85, 2016, [Online]. Available: <http://repository.its.ac.id/48842/>
- [16] Y. Yosefan Pane and J. J. Sihombing, "Bird Species Classification Using Transfer Learning Method," *J. Teknol. Terpadu*, vol. 9, no. 2, pp. 89–94, 2023.
- [17] G. Lukas Hansel and H. Bunyamin, "Penggunaan Augmentasi Data pada Klasifikasi Jenis Kanker Payudara dengan Model Resnet-34," *J. Strateg.*, vol. 3, no. 1, pp. 187–193, 2021.
- [18] S. Suprianto, D. S. Lestari, and O. H. Simung, "Aplikasi Penentuan Kesegaran Ikan Bandeng Menggunakan Metode Convolution Neural Network," *Insect (Informatics Secur. J. Tek. Inform.*, vol. 8, no. 2, pp. 77–86, 2023, doi: 10.33506/insect.v8i2.2196.
- [19] O. V. Putra, A. Musthafa, M. Nur, and M. Rido, "Classification of Calligraphy Writing Types Using Convolutional Neural Network Method (CNN)," *Procedia Eng. Life Sci.*, vol. 2, no. 1, pp. 2–8, 2021, doi: 10.21070/pels.v2i0.1136.
- [20] I. Edge *FOMO: Object detection for constrained devices*, <https://docs.edgeimpulse.com/docs/edge-impulse-studio/learning-blocks/object-detection/fomo-object-detection-for-constrained-devices>, Accessed: Apr. 21, 2024. [Online].