

PC Component Classification Using Convolutional Neural Network MobileNetV2 with Transfer Learning

Ilham Ismail^{1*}, Fauzi Nur Maqi², Nur Hidayah³

Universitas Nahdlatul Ulama Lampung

Corresponding Author: Ilham Ismail ilyakusuma99@gmail.com

ARTICLE INFO

Keywords : PC Components, Image Classification, MobileNetV2, Transfer Learning, Deep Learning

Received : 02 June 2025

Revised : 24 June 2025

Accepted: 26 July 2025

©2025 Ismail, Maqi, Hidayah:
This is an open-access article distributed under the terms of the [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/).



ABSTRACT

This research successfully developed and evaluated a computer vision-based classification system for PC components using the MobileNetV2 Convolutional Neural Network (CNN) architecture with transfer learning. A diverse dataset comprising 14 categories of PC components was collected and processed through stratified data splitting (70% training, 15% validation, 15% testing) and extensive data augmentation to enhance model robustness. The MobileNetV2 model, initialized with ImageNet pre-trained weights and fine-tuned with a custom classification layer, demonstrated effective convergence over 30 epochs. The model achieved a peak validation accuracy of 0.9143 and a final test accuracy of 92.38% with a loss of 0.2890. Performance analysis per class, using a classification report, indicated strong results, with most classes exhibiting precision, recall, and F1-scores above 0.90. Specifically, classes like keyboard, speakers, and webcam achieved perfect recall (1.00). While the confusion matrix revealed some misclassifications, particularly for 'case' and 'hdd' components, the overall performance is highly satisfactory, confirming the feasibility of MobileNetV2 for accurate PC component classification. This study contributes to automating PC component identification in industrial and technical contexts, paving the way for more efficient quality control, inventory management, and diagnostics.

Klasifikasi Komponen PC Menggunakan Convolutional Neural Network MobileNetV2 dengan Transfer Learning

Ilham Ismail^{1*}, Fauzi Nur Maqi², Nur Hidayah³

Universitas Nahdlatul Ulama Lampung

Corresponding Author: Ilham Ismail jyakusuma99@gmail.com

ARTICLE INFO

Kata Kunci: Komponen PC, Klasifikasi Gambar, MobileNetV2, Transfer Learning, Deep Learning

Received : 02 Juni 2025

Revised : 24 Juni 2025

Accepted: 26 Juli 2025

©2025 Ismail, Maqi, Hidayah:

This is an open-access article distributed under the terms of the [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/).



ABSTRAK

Penelitian ini berhasil mengembangkan dan mengevaluasi sistem klasifikasi komponen PC berbasis visi komputer menggunakan arsitektur Convolutional Neural Network (CNN) MobileNetV2 dengan teknik transfer learning. Dataset yang beragam, terdiri dari 14 kategori komponen PC, dikumpulkan dan diproses melalui pembagian data stratifikasi (70% pelatihan, 15% validasi, 15% pengujian) serta augmentasi data ekstensif untuk meningkatkan robustensi model. Model MobileNetV2, yang diinisialisasi dengan bobot pre-trained ImageNet dan disempurnakan dengan lapisan klasifikasi kustom, menunjukkan konvergensi yang efektif selama 30 epoch. Model mencapai puncak akurasi validasi sebesar 0.9143 dan akurasi pengujian akhir sebesar 92.38% dengan loss 0.2890. Analisis kinerja per kelas, menggunakan classification report, menunjukkan hasil yang kuat, dengan sebagian besar kelas menunjukkan precision, recall, dan F1-score di atas 0.90. Secara khusus, kelas seperti keyboard, speakers, dan webcam mencapai recall sempurna (1.00). Meskipun confusion matrix mengungkapkan beberapa misklasifikasi, terutama untuk komponen 'case' dan 'hdd', performa keseluruhan sangat memuaskan, memvalidasi kelayakan MobileNetV2 untuk klasifikasi komponen PC yang akurat. Studi ini berkontribusi pada otomatisasi identifikasi komponen PC dalam konteks industri dan teknis, membuka jalan bagi kontrol kualitas, manajemen inventaris, dan diagnostik yang lebih efisien.

PENDAHULUAN

Dalam era digital yang terus berkembang pesat, perangkat keras komputer (PC) menjadi tulang punggung berbagai aktivitas manusia, dari pekerjaan, komunikasi, hingga hiburan. Komponen-komponen penyusun PC, seperti Central Processing Unit (CPU), Graphics Processing Unit (GPU), Random Access Memory (RAM), dan motherboard, memiliki peran krusial dalam fungsionalitas sistem secara keseluruhan. Seiring dengan keragaman jenis dan model komponen yang terus meningkat, identifikasi dan klasifikasi komponen PC secara manual menjadi tugas yang menantang, memakan waktu, dan rentan terhadap kesalahan.

Kebutuhan akan identifikasi komponen PC secara otomatis menjadi semakin mendesak dalam berbagai aplikasi industri dan teknis. Misalnya, dalam proses manufaktur, identifikasi cepat dan akurat diperlukan untuk kontrol kualitas, inspeksi otomatis, atau perakitan [2]. Di sektor layanan, diagnostik PC yang efisien memerlukan pengenalan komponen yang cepat. Selain itu, manajemen inventaris yang akurat dan efisiensi dalam penanganan komponen dalam skala besar juga sangat diuntungkan dari sistem klasifikasi otomatis. Identifikasi otomatis ini dapat meningkatkan efisiensi dan mengurangi biaya operasional secara signifikan.

Identifikasi komponen PC secara manual, meskipun merupakan metode yang umum, sangat tidak efisien dan rentan terhadap kesalahan manusia, terutama mengingat jumlah dan variasi komponen yang besar. Tantangan lainnya meliputi kondisi visual yang bervariasi dari komponen (misalnya, sudut pandang, pencahayaan, bayangan, atau kerusakan fisik), yang dapat mempersulit pengenalan. Metode tradisional berbasis visi komputer yang ada seringkali menghadapi keterbatasan dalam menghadapi variasi kondisi dunia nyata ini, seperti perubahan perspektif, pencahayaan, dan buram akibat gerakan[1]. Oleh karena itu, diperlukan solusi yang lebih tangguh dan otomatis untuk mengklasifikasikan komponen PC dari gambar.

Penelitian ini mengusulkan penggunaan Jaringan Saraf Konvolusional (Convolutional Neural Network - CNN) sebagai pendekatan berbasis visi komputer untuk mengklasifikasikan berbagai komponen PC secara otomatis dari gambar. CNN telah merevolusi banyak area kecerdasan mesin, memungkinkan akurasi *superhuman* untuk tugas pengenalan gambar yang menantang

Secara spesifik, arsitektur MobileNetV2 dipilih sebagai *base model* dalam penelitian ini [3]. MobileNetV2 adalah model *deep neural network* yang dirancang khusus untuk lingkungan seluler dan terbatas sumber daya, yang mampu secara signifikan mengurangi jumlah operasi dan memori yang dibutuhkan sambil mempertahankan akurasi yang tinggi. Keunggulan ini menjadikan MobileNetV2 pilihan yang ideal untuk aplikasi klasifikasi yang membutuhkan efisiensi komputasi, seperti identifikasi komponen PC secara cepat.

Untuk mengatasi keterbatasan data dan meningkatkan performa model, teknik transfer learning akan diterapkan. *Transfer learning* adalah metodologi yang telah banyak digunakan dalam aplikasi *deep learning*, di mana lapisan-lapisan konvolusi dari model yang telah dilatih pada dataset besar (seperti

ImageNet) digunakan sebagai pengekstraksi fitur, sementara beberapa lapisan terakhir dilatih ulang sesuai dengan masalah yang dihadapi. Pendekatan ini memungkinkan model untuk mencapai akurasi klasifikasi yang tinggi bahkan dengan dataset yang relatif lebih kecil.

TINJAUAN PUSTAKA

Banyak penelitian telah dilakukan dalam mendeteksi dan mengklasifikasikan komponen elektronik menggunakan berbagai pendekatan kecerdasan buatan, terutama *deep learning*. Pendekatan ini relevan untuk berbagai tujuan, termasuk daur ulang limbah elektronik dan inspeksi kualitas.

Salah satu studi yang relevan, Fatih Uysal mengeksplorasi deteksi komponen elektronik menggunakan berbagai model *deep learning* berbasis jaringan saraf. Penelitian ini secara spesifik membandingkan lima model berbeda, termasuk MobileNetV2, EfficientNet, dan Vision Transformer (ViT), untuk deteksi lima jenis komponen elektronik (kapasitor, *chip*, LED, resistor, dan transistor). Studi ini melakukan augmentasi data pada dataset *open source* untuk meningkatkan jumlah data, dan melaporkan metrik evaluasi seperti *precision*, *recall*, *F1-score*, dan akurasi. Hasilnya menunjukkan akurasi tertinggi sebesar 0.992 pada model Data-efficient Image Transformer (DeiT), sementara MobileNetV2 mencapai akurasi 0.744.

Osmani et al. mengusulkan model *transfer learning* bernama VoltaVision untuk klasifikasi komponen elektronik, membandingkannya dengan model VGG dan ResNet, dan mencapai akurasi tertinggi 95.2% [6]. Hozyń juga menggunakan CNN untuk mengklasifikasikan 10 kelas komponen elektronik dalam aplikasi industri, mencapai akurasi 99.03% dengan model berbasis ResNet [7]. Penelitian lain oleh Surmeli dan Ekenel mengusulkan model Vision Transformer (ViT-Mini) untuk klasifikasi komponen PCB, mencapai akurasi 96.52% untuk klasifikasi multi-kelas enam kategori [8]. Ini menunjukkan bahwa CNN dan arsitektur berbasis *transformer* sangat efektif dalam tugas klasifikasi komponen elektronik.

Beberapa penelitian juga berfokus pada deteksi objek komponen elektronik pada Printed Circuit Board (PCB) menggunakan keluarga model You Only Look Once (YOLO) dan Single Shot MultiBox Detector (SSD). Misalnya, Varna dan Abromavicius menggunakan berbagai versi YOLO dan SSD-MobileNet untuk klasifikasi dan deteksi komponen (kapasitor, resistor, dioda, transistor), mencapai *Average Precision (AP)* tertinggi 94.08% pada YOLOv4 [9]. Huang et al. menggunakan YOLOv3 untuk deteksi empat jenis komponen elektronik dengan mAP 95.21% [10]. Penelitian terbaru seperti Mohsin et al. mencapai mAP 99% untuk deteksi komponen elektronik pada limbah PCB menggunakan YOLOv8, YOLOv9, dan RT-DETR [11]. Meskipun demikian, model YOLO dan SSD cenderung dioptimalkan untuk kecepatan, namun mungkin terbatas dalam mendeteksi objek kecil dan menentukan lokasi cacat secara tepat.

Transfer learning telah menjadi teknik yang sangat populer dalam *deep learning*, terutama untuk tugas klasifikasi gambar di mana dataset yang tersedia mungkin tidak cukup besar untuk melatih model dari awal. Konsep dasarnya adalah memanfaatkan fitur-fitur yang telah dipelajari oleh model yang dilatih

pada dataset yang sangat besar (misalnya, ImageNet) dan mengaplikasikan pengetahuan tersebut ke tugas baru yang terkait.

Razavian et al. melaporkan bahwa *deep CNN* dapat digunakan untuk klasifikasi dataset arbitrer dengan mengekstraksi fitur dari lapisan-lapisan konvolusi yang telah dilatih sebelumnya pada dataset lain yang besar. Metodologi ini memungkinkan lapisan konvolusi yang sudah mampu mengekstrak fitur kuat untuk dibekukan, sementara beberapa lapisan terakhir dapat dilatih ulang sesuai dengan masalah yang dihadapi. Pendekatan ini, yang disebut *transfer learning*, telah banyak digunakan dalam aplikasi *deep learning* dan terbukti secara signifikan meningkatkan akurasi, terutama untuk dataset terbatas .

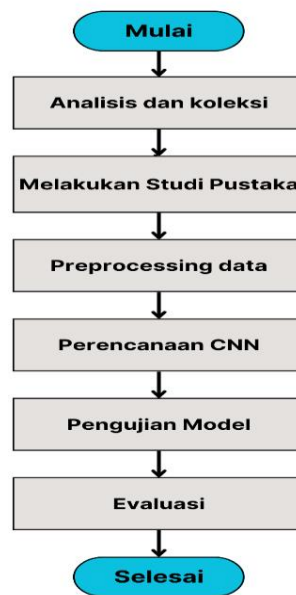
Dalam konteks klasifikasi MobileNetV2, *transfer learning* sering kali melibatkan pembekuan lapisan *base model* MobileNetV2 yang telah dilatih pada ImageNet, kemudian menambahkan dan melatih lapisan klasifikasi kustom di atasnya. Metode ini telah terbukti efektif dalam berbagai studi, termasuk klasifikasi gambar buah yang mencapai *precision*, *recall*, dan *F1-score* hingga 99%.

Banyak penelitian telah mengkonfirmasi efektivitas MobileNetV2 dalam berbagai tugas visi komputer, termasuk klasifikasi gambar, deteksi objek, dan segmentasi semantik, bahkan mengungguli model *real-time detector* lainnya dalam hal akurasi dan kompleksitas model dengan parameter yang jauh lebih sedikit. Fatih Uysal juga mengevaluasi MobileNetV2 dalam klasifikasi komponen elektronik, meskipun model berbasis *transformer* menunjukkan performa lebih tinggi dalam studi tersebut, MobileNetV2 tetap menjadi pilihan yang efisien .

METODE PENELITIAN

Studi ini menyajikan tinjauan terhadap penelitian-penelitian terdahulu yang relevan dengan topik klasifikasi komponen PC menggunakan metode visi komputer dan *deep learning*. Tinjauan ini mencakup eksplorasi berbagai arsitektur jaringan saraf konvolusional (CNN) yang telah diterapkan, peran penting teknik *transfer learning*, serta analisis performa yang dicapai dalam konteks identifikasi objek serupa atau komponen elektronik lainnya. Tujuan dari studi pustaka ini adalah untuk membangun landasan teoritis, memahami status terkini penelitian di bidang ini, mengidentifikasi berbagai pendekatan yang telah digunakan, serta menyoroti keunggulan dan keterbatasan dari metode-metode tersebut yang menjadi motivasi penelitian ini.

1. Alur Penelitian



Gambar 1. Alur Tahapan Penelitian

Proses metodologi penelitian ini, mulai dari tahap awal hingga akhir, ditunjukkan dalam diagram alir pada Gambar 3.1. Diagram ini memvisualisasikan urutan langkah-langkah yang sistematis, memastikan transparansi dan kejelasan dalam pelaksanaan studi.

2. Pengumpulan Data

Dataset yang digunakan dalam penelitian ini terdiri dari gambar-gambar komponen PC yang dikelompokkan ke dalam 14 kategori (kelas) yang berbeda. Gambar-gambar ini sebelumnya telah diorganisir dalam satu folder utama (pc_parts) di Google Drive, dengan masing-masing kategori memiliki subfolder terpisah.

Berikut adalah daftar 14 kategori komponen PC yang diklasifikasikan:

- cables
- case
- cpu
- gpu
- hdd
- headset
- keyboard
- microphone
- monitor
- motherboard
- mouse
- ram
- speakers
- webcam

Total gambar yang terkumpul adalah 2800 gambar. Variasi gambar yang berusaha dikumpulkan meliputi berbagai sudut pandang, kondisi pencahayaan,

latar belakang, serta jenis atau merek komponen yang berbeda dalam setiap kategori. Hal ini bertujuan untuk meningkatkan robustensi model terhadap variasi gambar di dunia nyata.

3. Pra-pemrosesan Data

Sebelum dapat digunakan untuk pelatihan model *deep learning*, data gambar perlu melalui tahap pra-pemrosesan. Tahap ini memastikan data dalam format yang sesuai dan mengoptimalkan proses pembelajaran model.

1. Pembagian Dataset: Dataset asli yang berada dalam satu folder utama dibagi secara programatis menjadi tiga subset: pelatihan (training), validasi (validation), dan pengujian (testing). Pembagian ini dilakukan dengan rasio sebagai berikut:
 - o Data Pelatihan (Training Data): 70% dari total dataset (sebanyak 1960 gambar).
 - o Data Validasi (Validation Data): 15% dari total dataset (sebanyak 420 gambar).
 - o Data Pengujian (Testing Data): 15% dari total dataset (sebanyak 420 gambar). Pembagian dataset dilakukan secara stratified berdasarkan label kelas, memastikan bahwa distribusi jumlah gambar untuk setiap kelas tetap proporsional di setiap subset. Pengaturan `random_state=42` digunakan untuk menjamin reproduktibilitas hasil pembagian data.
2. Augmentasi dan Normalisasi Data: Untuk memperkaya variasi data pelatihan dan mencegah *overfitting*, digunakan teknik augmentasi data menggunakan ImageDataGenerator dari Keras. Augmentasi yang diterapkan pada data pelatihan meliputi:
 - o Rotasi acak (`rotation_range=20`)
 - o Pergeseran lebar horizontal acak (`width_shift_range=0.2`)
 - o Pergeseran tinggi vertikal acak (`height_shift_range=0.2`)
 - o Transformasi *shear* acak (`shear_range=0.2`)
 - o *Zoom in/out* acak (`zoom_range=0.2`)
 - o Pembalikan horizontal (`horizontal_flip=True`)
 - o Pengisian piksel yang hilang dengan mode nearest (`fill_mode='nearest'`)

Semua gambar, baik untuk pelatihan, validasi, maupun pengujian, diubah ukurannya menjadi 224x224 piksel. Ukuran ini adalah dimensi input standar yang dibutuhkan oleh arsitektur MobileNetV2 yang telah dilatih pada dataset ImageNet.

Normalisasi nilai piksel dilakukan menggunakan fungsi `preprocess_input` dari `tensorflow.keras.applications.mobilenet_v2`. Fungsi ini mengubah rentang nilai piksel dari `[0, 255]` menjadi `[-1, 1]`, yang sesuai dengan pra-pemrosesan yang digunakan selama pelatihan MobileNetV2 di ImageNet. Untuk data validasi dan pengujian, hanya normalisasi ini yang diterapkan tanpa augmentasi.

Data dimuat dalam *batches* dengan ukuran `BATCH_SIZE = 32` untuk efisiensi komputasi. Label kelas dikodekan dalam format *one-hot encoding* menggunakan `class_mode='categorical'`.

4. Arsitektur Model

Penelitian ini mengadopsi pendekatan *transfer learning* dengan memanfaatkan arsitektur MobileNetV2 sebagai *base model*. MobileNetV2 adalah arsitektur CNN yang efisien secara komputasi, dirancang untuk perangkat dengan sumber daya terbatas, namun tetap menawarkan akurasi yang kompetitif.

Model MobileNetV2 dimuat dengan bobot yang telah dilatih sebelumnya pada dataset ImageNet (`weights='imagenet'`). Lapisan klasifikasi *top* dari MobileNetV2 yang asli tidak disertakan (`include_top=False`). Hal ini memungkinkan penambahan lapisan klasifikasi kustom yang disesuaikan dengan 14 kelas komponen PC yang menjadi target penelitian.

Model yang dibangun terdiri dari:

- Lapisan Input: Menerima gambar dengan bentuk (224, 224, 3) (tinggi, lebar, kanal RGB).
- MobileNetV2 Base Model: Lapisan-lapisan konvolusional dari MobileNetV2 yang berfungsi sebagai pengekstraksi fitur. Bobot dari lapisan ini dibekukan (`base_model.trainable = False`) pada tahap awal pelatihan. Pembekuan ini bertujuan untuk memanfaatkan pengetahuan yang telah diperoleh dari dataset ImageNet, sekaligus mengurangi jumlah parameter yang perlu dilatih, mempercepat konvergensi, dan mencegah *overfitting* pada dataset yang relatif lebih kecil.
- Lapisan GlobalAveragePooling2D: Lapisan ini ditempatkan di atas *base model* untuk meratakan peta fitur spasial menjadi vektor tunggal.
- Lapisan Output (Dense): Lapisan *Dense* dengan 14 neuron (sesuai dengan jumlah kelas) dan fungsi aktivasi softmax. Fungsi softmax menghasilkan distribusi probabilitas untuk setiap kelas, menunjukkan kemungkinan gambar termasuk dalam kategori tertentu.

Ringkasan model menunjukkan bahwa model memiliki 2,275,918 total parameter, di mana hanya 17,934 parameter yang dapat dilatih (dari lapisan kustom yang ditambahkan), sementara 2,257,984 parameter lainnya adalah non-trainable (dari *base model* MobileNetV2 yang dibekukan).

5. Konfigurasi Pelatihan Model

Model dikompilasi dengan konfigurasi berikut:

- Optimizer: `tf.keras.optimizers.Adam` digunakan sebagai optimizer, yang merupakan pilihan populer karena efisiensinya dalam mengelola *learning rate* secara adaptif.
- Learning Rate: Tingkat pembelajaran awal diatur ke 0.0001. Nilai yang kecil ini umumnya direkomendasikan untuk transfer learning agar bobot *pretrained* tidak terlalu banyak berubah secara drastis pada tahap awal pelatihan.

- Fungsi Kerugian (Loss Function): `categorical_crossentropy` digunakan sebagai fungsi kerugian, yang cocok untuk masalah klasifikasi multi-kelas dengan label *one-hot encoded*.
- Metrik Evaluasi: `accuracy` digunakan sebagai metrik utama untuk memantau performa model selama pelatihan.

Proses pelatihan model dilakukan selama 30 epoch. Untuk mengoptimalkan proses pelatihan dan mencegah *overfitting*, dua *callback* utama diimplementasikan:

- `ModelCheckpoint`: `Callback` ini memantau akurasi pada set validasi (`val_accuracy`) dan secara otomatis menyimpan bobot model terbaik (yaitu model dengan `val_accuracy` tertinggi) ke file `/content/best_model_v2.h5`. Ini memastikan bahwa versi model dengan performa generalisasi terbaik tetap tersimpan.
- `EarlyStopping`: `Callback` ini memantau nilai `val_loss` dengan *patience* 7. Jika `val_loss` tidak menunjukkan penurunan selama 7 epoch berturut-turut, pelatihan akan dihentikan secara otomatis. Setelah penghentian, bobot model akan dikembalikan ke epoch terbaik yang ditemukan selama pelatihan. Ini membantu mencegah model dari *overfitting* yang berlebihan pada data pelatihan.

HASIL DAN PEMBAHASAN

1. Hasil Pelatihan Model

Proses pelatihan model `MobileNetV2` dilakukan dalam dua tahap: pertama selama 10 *epoch*, kemudian dilanjutkan dengan 30 *epoch* tambahan. Pada tahap awal, model menunjukkan peningkatan akurasi yang cepat pada kedua set data pelatihan dan validasi.

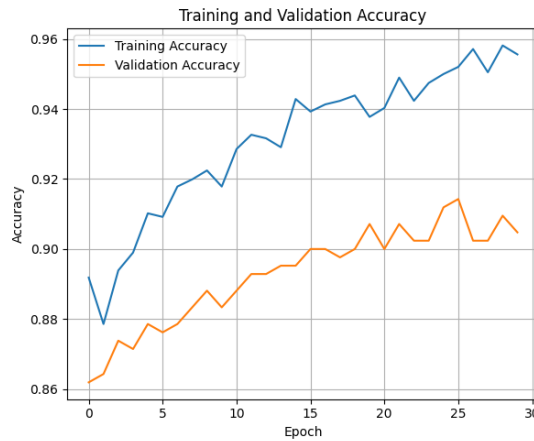
Pada akhir pelatihan 10 *epoch* pertama:

- Akurasi pelatihan (`Training Accuracy`) mencapai 0.8820.
- `Loss` pelatihan (`Training Loss`) adalah 0.5932.
- Akurasi validasi (`Validation Accuracy`) mencapai 0.8500.
- `Loss` validasi (`Validation Loss`) adalah 0.6203.

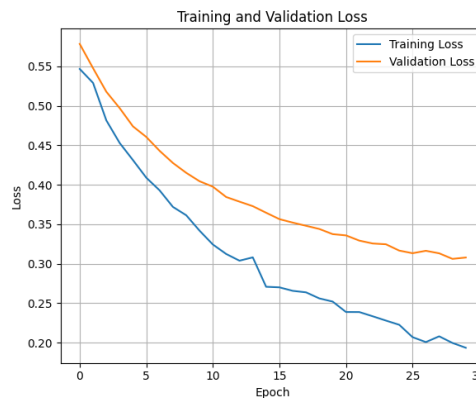
Peningkatan akurasi validasi dari 0.2190 pada *epoch* 1 menjadi 0.8500 pada *epoch* 10 menunjukkan bahwa model berhasil mempelajari fitur-fitur penting dari dataset komponen PC. `Callback ModelCheckpoint` berhasil menyimpan model terbaik pada *epoch* 10 karena `val_accuracy` terus meningkat hingga titik tersebut.

Melihat potensi peningkatan lebih lanjut, pelatihan dilanjutkan untuk 30 *epoch* tambahan dengan menggunakan `callback` `EarlyStopping` yang memiliki *patience* 7. Selama pelatihan lanjutan ini, model terus menunjukkan peningkatan performa. Puncak akurasi validasi tercatat pada Epoch 26 dari sesi pelatihan lanjutan (atau total sekitar *epoch* 36 jika dihitung dari awal), di mana `val_accuracy` mencapai 0.9143 dan `val_loss` menurun hingga 0.3132. Ini menunjukkan bahwa model berhasil mencapai tingkat generalisasi yang tinggi, melampaui target akurasi 90%.

Grafik di bawah ini menggambarkan perubahan akurasi dan `loss` pada set data pelatihan dan validasi selama seluruh proses pelatihan.



Gambar 2. Grafik Akurasi Pelatihan dan Validasi



Gambar 3. Grafik Loss Pelatihan dan Validasi

Dari grafik, terlihat bahwa kurva akurasi pelatihan dan validasi cenderung naik secara bersamaan, sementara kurva *loss* pelatihan dan validasi cenderung menurun. Meskipun ada sedikit jarak (*gap*) antara akurasi pelatihan dan validasi pada *epoch-epoch* akhir (akurasi pelatihan mencapai sekitar 0.9560 sedangkan validasi di 0.9143), ini merupakan indikasi normal dari *overfitting* ringan yang berhasil dikelola dengan baik oleh teknik augmentasi data dan *EarlyStopping*. *EarlyStopping* memastikan model tidak terus berlatih hingga performa pada data yang belum terlihat mulai memburuk, sehingga bobot terbaik dapat dipulihkan.

2. Evaluasi Model pada Data Pengujian

Setelah pelatihan selesai, model terbaik yang disimpan (*best_model_v2.h5*) dimuat dan dievaluasi secara independen pada set data pengujian yang sama sekali belum pernah dilihat oleh model. Evaluasi ini memberikan estimasi yang tidak bias mengenai performa model di skenario dunia nyata.

Hasil evaluasi pada *test set* adalah sebagai berikut:

- Test Loss: 0.2890.
- Test Accuracy: 92.38%.

Akurasi sebesar 92.38% pada *test set* merupakan hasil yang sangat memuaskan, mengkonfirmasi kemampuan model untuk mengklasifikasikan komponen PC dengan tingkat keandalan yang tinggi.

3. Analisis Kinerja Per Kelas

Untuk mendapatkan pemahaman yang lebih mendalam mengenai performa model untuk setiap kategori komponen PC, analisis dilakukan menggunakan *Classification Report* yang mencakup metrik *Precision*, *Recall*, dan *F1-score*.

Tabel 1. Laporan Klasifikasi Model pada Data Pengujian

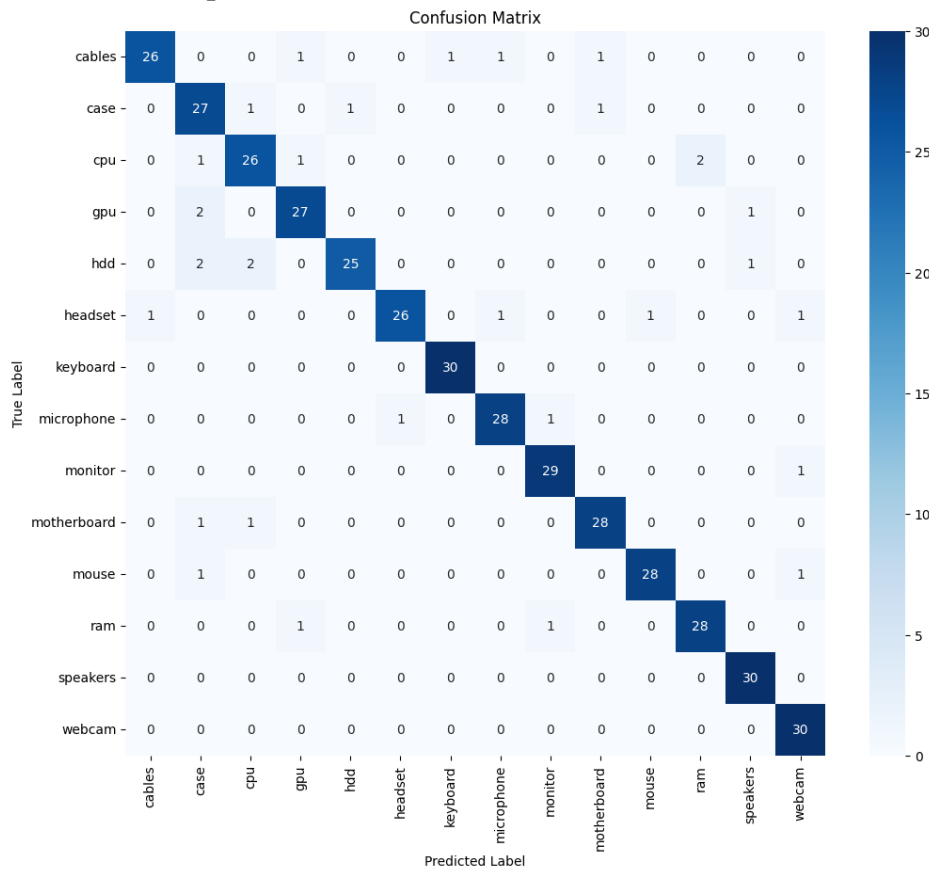
Kelas	Precision	Recall	F1-score	Support
cables	0.96	0.87	0.91	30
case	0.79	0.90	0.84	30
cpu	0.87	0.87	0.87	30
gpu	0.90	0.90	0.90	30
hdd	0.96	0.83	0.89	30
headset	0.96	0.87	0.91	30
keyboard	0.97	1.00	0.98	30
microphone	0.93	0.93	0.93	30
monitor	0.94	0.97	0.95	30
motherboard	0.93	0.93	0.93	30
mouse	0.97	0.93	0.95	30
ram	0.93	0.93	0.93	30
speakers	0.94	1.00	0.97	30
webcam	0.91	1.00	0.95	30
accuracy			0.92	420
macro avg	0.93	0.92	0.92	420
weighted avg	0.93	0.92	0.92	420

Dari Tabel 1, dapat diamati bahwa:

- Kinerja Unggul: Kelas-kelas seperti keyboard, speakers, dan webcam menunjukkan *recall* sempurna (1.00), yang berarti model berhasil mengidentifikasi semua sampel dari kelas tersebut dengan benar. Selain itu, *precision* dan *F1-score* untuk kelas-kelas ini juga sangat tinggi, mendekati 0.97-0.98, menunjukkan bahwa prediksi model untuk kelas-kelas ini sangat akurat.
- Kinerja Sangat Baik: Mayoritas kelas lainnya seperti cables, cpu, gpu, headset, microphone, monitor, motherboard, mouse, dan ram menunjukkan *precision*, *recall*, dan *F1-score* di atas 0.90 atau mendekati 0.90, mengindikasikan performa klasifikasi yang sangat kuat.
- Kelas yang Lebih Menantang: Kelas case menunjukkan *precision* terendah (0.79), meskipun *recall*-nya cukup baik (0.90). Ini berarti model terkadang salah mengklasifikasikan komponen lain sebagai case. Sementara itu, kelas hdd memiliki *precision* yang tinggi (0.96) tetapi *recall* yang relatif rendah (0.83), mengindikasikan bahwa model cenderung melewatkan beberapa sampel hdd atau mengklasifikasikannya ke kelas lain. Perbedaan performa ini mungkin disebabkan oleh variasi visual yang lebih besar atau kemiripan fitur dengan komponen lain dalam dataset.

4. Analisis Confusion Matrix

Confusion matrix memberikan rincian visual mengenai prediksi benar dan salah untuk setiap kelas.



Gambar 4. Confusion Matrix Model pada Data Pengujian

Dari *confusion matrix* (Gambar 3), angka pada diagonal utama merepresentasikan jumlah prediksi yang benar. Angka di luar diagonal menunjukkan misklasifikasi.

Contoh Misklasifikasi:

- Dari 30 sampel case yang sebenarnya, 27 diklasifikasikan dengan benar, namun 1 sampel salah diklasifikasikan sebagai cpu, 1 sebagai hdd, dan 1 sebagai motherboard.
- Untuk kelas hdd, 25 dari 30 sampel diklasifikasikan dengan benar. Namun, ada 2 sampel hdd yang salah diklasifikasikan sebagai case, 2 sebagai cpu, dan 1 sebagai speakers.
- Kelas headset memiliki 1 sampel yang salah diklasifikasikan sebagai cables, 1 sebagai microphone, dan 1 sebagai mouse.

Kesalahan klasifikasi ini mungkin disebabkan oleh beberapa faktor, seperti kemiripan visual antar komponen tertentu (misalnya, beberapa kabel mungkin sulit dibedakan jika tidak ada konteks perangkatnya), variasi pencahayaan atau sudut pengambilan gambar yang ekstrem pada data pelatihan, atau representasi fitur yang kurang kuat untuk membedakan antara kelas-kelas yang sulit tersebut. Meskipun demikian, secara keseluruhan, pola diagonal yang kuat menegaskan kemampuan model yang baik dalam klasifikasi.

KESIMPULAN DAN SARAN

Penelitian ini telah berhasil mengembangkan dan mengevaluasi sistem klasifikasi komponen PC berbasis visi komputer menggunakan arsitektur *Convolutional Neural Network* (CNN) MobileNetV2 dengan teknik *transfer learning*. Dataset yang terdiri dari 14 kategori komponen PC yang beragam berhasil dikumpulkan dan diproses melalui tahapan pra-pemrosesan yang meliputi pembagian data stratifikasi (70% pelatihan, 15% validasi, 15% pengujian) dan augmentasi data ekstensif untuk meningkatkan robustensi model.

Model MobileNetV2, yang diinisialisasi dengan bobot *pre-trained* ImageNet dan dibekukan pada lapisan *base model*, kemudian dilatih ulang dengan lapisan klasifikasi kustom. Proses pelatihan selama 30 *epoch* dengan *optimizer* Adam dan *learning rate* 0.0001, serta didukung oleh *callbacks* ModelCheckpoint dan EarlyStopping, menunjukkan konvergensi yang efektif. Model mencapai puncak akurasi validasi sebesar 0.9143 pada *epoch* 26 dan akurasi pengujian akhir sebesar 92.38% dengan *loss* 0.2890.

Analisis kinerja per kelas melalui *classification report* menunjukkan bahwa model berkinerja sangat kuat, dengan sebagian besar kelas mencapai *precision*, *recall*, dan *F1-score* di atas 0.90. Kelas seperti keyboard, speakers, dan webcam bahkan mencapai *recall* sempurna (1.00). Meskipun demikian, *confusion matrix* mengungkapkan adanya misklasifikasi pada beberapa kelas, seperti case yang kadang tertukar dengan cpu atau hdd, dan hdd yang kadang salah diklasifikasikan sebagai case atau cpu. Hal ini mengindikasikan adanya kemiripan visual antar komponen tertentu atau variasi gambar yang menantang, namun secara keseluruhan, performa model sangat memuaskan dan menegaskan kelayakan penggunaan MobileNetV2 untuk tugas klasifikasi komponen PC.

Saran

Berdasarkan temuan dalam penelitian ini, beberapa arah pengembangan dapat dipertimbangkan untuk pekerjaan di masa depan:

1. Ekspansi Dataset: Menambah jumlah dan variasi gambar pada dataset, terutama untuk kelas-kelas yang menunjukkan performa sedikit lebih rendah atau memiliki pola misklasifikasi tertentu, dapat lebih meningkatkan akurasi dan generalisasi model.
2. Fine-tuning Lanjutan: Mengeksplorasi strategi *fine-tuning* yang lebih agresif, seperti membuka (*unfreeze*) dan melatih ulang beberapa lapisan terakhir dari *base model* MobileNetV2 dengan *learning rate* yang sangat kecil, berpotensi memeras akurasi tambahan.
3. Eksplorasi Arsitektur Alternatif/Hibrida: Membandingkan MobileNetV2 dengan arsitektur model efisien lainnya (misalnya, EfficientNet, MobileNetV3) atau bahkan model *object detection* (misalnya, YOLO, Faster R-CNN) yang tidak hanya mengklasifikasikan tetapi juga melokalisasi komponen, dapat memberikan solusi yang lebih komprehensif.
4. Penanganan Ketidakseimbangan Kelas: Jika ada ketidakseimbangan jumlah gambar antar kelas, teknik penanganan *class imbalance* (misalnya,

oversampling, undersampling, weighted loss) dapat diterapkan untuk memastikan model belajar secara adil dari semua kelas.

5. Validasi di Lingkungan Nyata: Mengimplementasikan dan menguji model dalam skenario aplikasi industri yang sesungguhnya untuk memvalidasi performa dalam kondisi operasional yang lebih kompleks.

PENELITIAN LANJUTAN

Berdasarkan hasil yang telah dicapai, penelitian lanjutan dapat difokuskan pada peningkatan akurasi dan generalisasi model melalui eksplorasi arsitektur deep learning yang lebih kompleks atau hybrid, seperti EfficientNet, Vision Transformer (ViT), atau kombinasi CNN dengan attention mechanism. Selain itu, penambahan dataset dengan variasi pencahayaan, latar belakang, dan sudut pengambilan gambar yang lebih beragam dapat membantu mengurangi tingkat misklasifikasi, terutama pada kelas-kelas yang memiliki kemiripan visual tinggi. Penelitian juga dapat menguji performa model dalam kondisi dunia nyata (*real-time inference*) dengan perangkat edge seperti Raspberry Pi atau Jetson Nano untuk menilai efisiensi dan skalabilitas sistem dalam konteks aplikasi praktis.

Aspek interpretabilitas model juga menjadi arah penting untuk penelitian berikutnya, seperti penerapan Grad-CAM atau teknik visualisasi lain guna memahami area fokus model saat melakukan prediksi. Pendekatan ini dapat membantu mengidentifikasi potensi bias atau fitur visual yang membingungkan model. Selain itu, perlu dilakukan studi usability dan integrasi sistem ini dalam lingkungan pendidikan atau layanan teknis untuk mengukur dampak nyata dan penerimaan pengguna. Dengan pengembangan lebih lanjut, sistem klasifikasi komponen PC berbasis visi komputer ini berpotensi besar untuk diterapkan dalam berbagai bidang, mulai dari otomasi inventarisasi hingga pelatihan teknisi di bidang teknologi informasi.

UCAPAN TERIMA KASIH

Kami berterima kasih kepada Allah SWT karena telah memberi kami kekuatan dan bimbingan untuk menyelesaikan penelitian ini. Saya ingin mengucapkan terima kasih kepada orang tua yang telah memberikan dukungan dan doa tanpa syarat dan teman-teman saya yang telah memberikan semangat dan dukungan dalam menyusun penelitian ini.

DAFTAR PUSTAKA

- I. A. Soomro, A. Ahmad, dan R. H. Raza, "Printed Circuit Board identification using Deep Convolutional Neural Networks to facilitate recycling," *Resour Conserv Recycl*, vol. 177, Feb 2022, doi: 10.1016/j.resconrec.2021.105963.
- K. D. Weerakkody, R. Balasundaram, E. Osagie, dan J. Alshehabi Al-Ani, "Automated Defect Identification System in Printed Circuit Boards Using Region-Based Convolutional Neural Networks," *Electronics (Switzerland)*, vol. 14, no. 8, Apr 2025, doi: 10.3390/electronics14081542.
- M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, dan L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," 2018. doi: 10.48550/arXiv.1704.04861.
- M. Andreetto Google Inc dan A. G. Howard Menglong Zhu Bo Chen Dmitry Kalenichenko Weijun Wang Tobias Weyand Marco Andreetto, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," 2017, doi: 10.48550/arXiv.1704.04861.
- F. Uysal, "Electronic Components Detection Using Various Deep Learning Based Neural Network Models," *International Journal of Computational and Experimental Science and Engineering*, vol. 11, no. 1, hlm. 542-549, 2024, doi: 10.22399/ijcesen.855.
- A. Osmani, T. Rahman, dan S. Islam, *Voltavision: A Transfer Learning Model For Electronic Component Classification*. 2024.
- S. Hożyń, "Convolutional Neural Networks for Classifying Electronic Components in Industrial Applications," *Energies (Basel)*, vol. 16, no. 2, Jan 2023, doi: 10.3390/en16020887.
- C. Surmeli dan H. Ekenel, *An Efficient Vision Transformer Model for PCB Component Classification*. 2023. doi: 10.1109/SIU59756.2023.10224039.
- D. Varna dan V. Abromavičius, "A System for a Real-Time Electronic Component Detection and Classification on a Conveyor Belt," *Applied Sciences (Switzerland)*, vol. 12, no. 11, Jun 2022, doi: 10.3390/app12115608.
- R. Huang, J. Gu, X. Sun, H. Yongtao, dan S. Uddin, "A Rapid Recognition Method for Electronic Components Based on the Improved YOLO-V3 Network,"

Electronics (Basel), vol. 8, hlm. 825, Jul 2019, doi: 10.3390/electronics8080825.

M. Mohsin, S. Rovetta, F. Masulli, dan A. Cabri, *Real-Time Detection of Electronic Components in Waste Printed Circuit Boards: A Transformer-Based Approach*. 2024. doi: 10.48550/arXiv.2409.16496.

A. S. Razavian, H. Azizpour, J. Sullivan, dan S. Carlsson, "CNN Features Off-the-Shelf: An Astounding Baseline for Recognition," dalam *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, hlm. 512–519. doi: 10.1109/CVPRW.2014.131.