

IMPLEMENTASI CONVOLUTIONAL NEURAL NETWORK (CNN) UNTUK INSPEKSI CACAT PADA BENANG

¹Ana Mariam, ²Emmanuel Agung Nugroho*, ³Slamet Riyadi

^{1,2,3}Teknologi Rekayasa Mekatronika, Politeknik Enjineri Indorama, Purwakarta, Indonesia
e-mail: ¹annamariam971@gmail.com, ²emmanuel.agung@pei.ac.id, ³slamet@pei.ac.id.

Abstrak

PT Indorama Synthetics Tbk merupakan salah satu produsen tekstil yang terus berinovasi untuk mengembangkan sistem inspeksi berbasis teknologi computer vision saat ini. Selama ini teknologi computer vision telah digunakan untuk mengidentifikasi corak cone tip pada benang spinning berdasarkan kesamaan spesifikasi jenis benang. Penelitian ini dilakukan sebagai pengembangan sistem tersebut sehingga memiliki kemampuan untuk mengidentifikasi dan mengklasifikasikan hasil produksi berdasarkan kategori benang bagus dan benang cacat seperti Stripe, Dirty, dan Moldy. Tujuan penelitian ini adalah menghasilkan alat yang mampu melakukan inspeksi secara otomatis, sehingga meningkatkan efisiensi proses inspeksi produksi benang yang masih bergantung pada pengamatan manual. Teknologi yang dikembangkan pada penelitian ini mengintegrasikan computer vision dengan machine learning berbasis algoritma Convolutional Neural Network yang menghasilkan model Tensor Flow Lite yang di deploy kedalam Raspberry Pi. Pengujian model menggunakan dataset training, dan dianalisis berdasarkan akurasi, loss, serta performa pemrosesan citra secara real-time. Rata-rata hasil inferensi real-time menunjukkan nilai 68.13 ms untuk waktu inferensi dengan FPS 8.8 FPS dan akurasi 82.88%. Rata-rata Training Loss dengan nilai 0.9, Validation Loss 0.94, Training Accuracy 0.59, dan Validation Accuracy 0.56.

Kata kunci: Computer Vision; Inspeksi Benang; Convolutional Neural Network; TensorFlow Lite; Raspberry Pi.

Abstract

PT Indorama Synthetics Tbk is one of the leading textile manufacturers that continues to innovate in developing a computer vision-based inspection system. Currently, computer vision technology has been used to identify cone tip patterns in spinning yarn based on the similarity of yarn specifications. This research was conducted to further develop the system, enabling it to identify and classify production outcomes into categories of good-quality yarn and defective yarn, such as Stripe, Dirty, and Moldy. The objective of this study is to develop a tool capable of performing automatic inspections, thereby increasing the efficiency of the yarn inspection process, which still relies on manual observation. The technology developed in this research integrates computer vision with machine learning using a Convolutional Neural Network algorithm, resulting in a TensorFlow Lite model deployed on a Raspberry Pi. The model was tested using a training dataset and analyzed based on accuracy, loss, and real-time image processing performance. The real-time inference results show an average inference time of 68.13 ms, an FPS of 8.8, and an accuracy of 82.88%. The average Training Loss is 0.9, Validation Loss is 0.94, Training Accuracy is 0.59, and Validation Accuracy is 0.56.

Keywords: Computer Vision; Yarn Inspection; Convolutional Neural Network; TensorFlow Lite; Raspberry Pi

1. PENDAHULUAN

Dalam industri tekstil, teknologi pengolahan citra sudah banyak diterapkan khususnya dalam proses inspeksi untuk meningkatkan akurasi identifikasi kualitas objek dibandingkan dengan sensor konvensional yang hanya mampu mendeteksi keberadaan objek [1]. Pendekatan ini memungkinkan proses inspeksi otomatis yang lebih presisi, sehingga dapat mengurangi ketergantungan pada inspeksi manual. Namun, dalam implementasinya, sistem inspeksi otomatis yang digunakan masih memiliki beberapa keterbatasan. Misalnya, PT Indorama Synthetics Tbk merupakan salah satu produsen tekstil yang saat ini mengandalkan sistem computer vision untuk mengidentifikasi corak cone tip pada benang spinning. Namun, sistem tersebut belum memiliki kemampuan untuk secara otomatis mengidentifikasi dan mengklasifikasikan cacat benang seperti Stripe, Dirty, dan Mold. Akibatnya, proses inspeksi cacat benang masih bergantung pada pengamatan manual (human vision), yang berpotensi mengurangi efisiensi serta meningkatkan risiko kesalahan dalam identifikasi cacat benang.

Berangkat dari permasalahan tersebut, penelitian ini mengembangkan sistem pengolahan citra berbasis Convolutional Neural Network (CNN) untuk mendeteksi dan mengklasifikasikan cacat benang secara otomatis. CNN dipilih karena kemampuannya dalam mengekstraksi fitur secara otomatis, mengenali pola kompleks, serta memiliki akurasi tinggi dalam klasifikasi gambar [2].

Penelitian sebelumnya oleh Qur'ana (2023) telah menunjukkan efektivitas CNN dalam klasifikasi motif pada kain Sasirangan, warisan budaya khas suku Banjar di Kalimantan Selatan. Dalam penelitian tersebut, model CNN yang digunakan mampu mencapai akurasi sebesar 82% pada data latih dan 76% pada data uji, setelah dilakukan augmentasi untuk meningkatkan jumlah data [3]. Hasil ini menunjukkan bahwa CNN dapat menjadi metode yang andal dalam klasifikasi citra pada industri tekstil.

Penelitian lain yang dilakukan oleh Duanti et al. (2025) menunjukkan keberhasilan implementasi model Xception berbasis CNN dalam klasifikasi kanker kulit dengan akurasi mencapai 85%-99% setelah dioptimalkan menggunakan TensorFlow Lite pada perangkat Raspberry Pi [4].

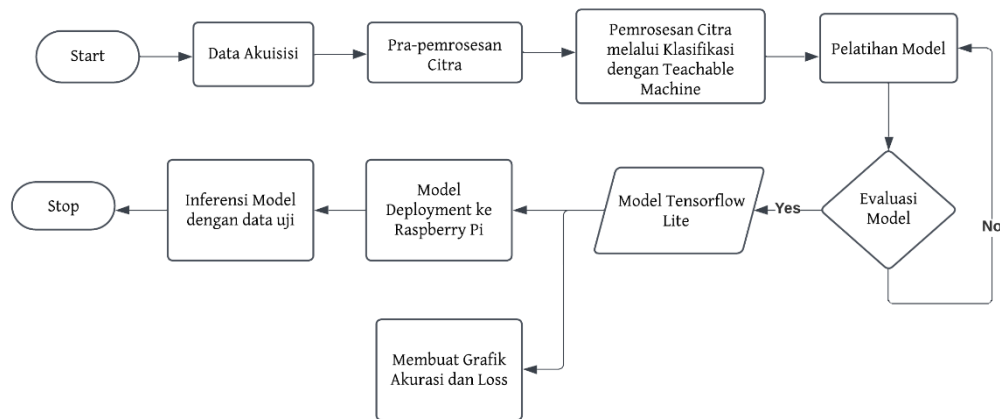
Prabowo et al. (2024) mengembangkan sistem deteksi warna kulit berbasis Convolutional Neural Network (CNN) melalui Teachable Machine Learning untuk menentukan kecocokan warna pakaian dengan warna kulit individu. Penelitian ini melibatkan pengumpulan citra kulit, preprocessing menggunakan model warna YCbCr, dan pelatihan CNN. Hasilnya menunjukkan akurasi tinggi dalam klasifikasi warna kulit dan rekomendasi warna pakaian yang sesuai, dengan potensi aplikasi dalam industri fashion dan personalisasi pakaian [5].

Sejalan dengan penelitian-penelitian sebelumnya, penelitian ini juga mengembangkan sistem klasifikasi berbasis Convolutional Neural Network (CNN) yang dikonversi ke TensorFlow Lite dari platform Teachable Machine untuk memungkinkan inferensi pada perangkat Raspberry Pi. Sistem ini dirancang untuk mengklasifikasikan empat kategori benang dengan jumlah produksi terbesar di PT Indorama Synthetics Tbk, yaitu Normal, Stripe, Mold, dan Dirty. Dengan demikian, sistem ini diharapkan dapat meningkatkan akurasi deteksi serta efisiensi dalam proses inspeksi benang.

Convolutional Neural Network (CNN) merupakan bentuk pengembangan dari Multilayer Perceptron (MLP) yang dirancang untuk memproses data dua dimensi [6]. Dengan struktur jaringan yang terdiri dari beberapa lapisan, CNN termasuk dalam kategori Deep Neural Network. Arsitektur ini pertama kali diperkenalkan oleh Kuniyiko Fukushima, seorang peneliti dari NHK Broadcasting Science Research Laboratories, dan kemudian dikembangkan lebih lanjut oleh Yann LeCun dalam penelitiannya [7]. CNN merupakan bagian dari Deep Neural Network yang mengadopsi prinsip kerja sistem saraf manusia, di mana informasi visual yang diterima diproses dan disimpan dalam bentuk peta topografi memori di otak. Setiap lapisan dalam CNN memiliki peran spesifik dalam mengenali dan mengekstraksi karakteristik suatu objek [8]. Perkembangan CNN sebagai bagian dari Deep Neural Network telah mendorong berbagai penelitian dalam penerapannya pada berbagai bidang, terutama dalam pengenalan pola dan klasifikasi gambar.

2. METODE PENELITIAN

Penelitian ini berfokus pada pengembangan dan penerapan sistem klasifikasi jenis benang berdasarkan cacatnya dengan memanfaatkan model TensorFlow Lite. Sistem ini dirancang untuk berjalan pada perangkat Raspberry Pi dengan Webcam sebagai alat pemrosesan citra. Rincian tahapan penelitian akan dijelaskan lebih lanjut pada bagian berikut dan dapat dilihat pada Gambar 1.

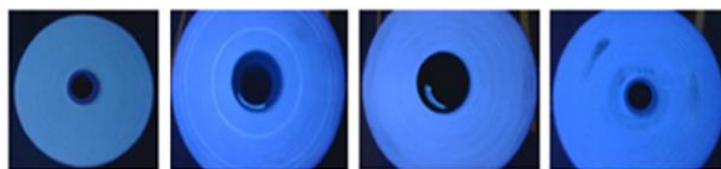


Gambar 1. Flowchart Metodologi Penelitian.

Gambar 1 menunjukkan Flowchart yang menggambarkan tahapan pemrosesan data dan implementasi model klasifikasi berbasis TensorFlow Lite pada Raspberry Pi. Proses dimulai dari akuisisi data, diikuti oleh pra-pemrosesan citra dan klasifikasi menggunakan Teachable Machine untuk melatih model. Setelah melalui tahap pelatihan dan evaluasi, model yang telah dikonversi ke format TensorFlow Lite digunakan untuk dua tujuan utama, yaitu deployment ke Raspberry Pi untuk inferensi dengan data uji serta pembuatan grafik akurasi dan loss guna mengevaluasi performa model menggunakan data training dan validasi.

2.1 Data Akuisisi

Dalam pemrosesan citra digital, akuisisi citra merujuk pada tahapan pengambilan dan pengumpulan gambar dari berbagai sumber, seperti kamera untuk dianalisis lebih lanjut. Proses ini mencakup transformasi citra analog ke dalam format digital agar dapat diolah oleh komputer. Akuisisi citra juga bisa didefinisikan sebagai tahap pengambilan gambar yang digunakan sebagai dataset dalam penelitian [9]. Pada tahapan data akuisisi ini, proses dimulai dari pemilihan benang cacat dan benang normal. Dimana benang yang digunakan merupakan jenis benang spinning dengan berbagai kategori yaitu benang normal, dan benang cacat yang terbagi menjadi benang belang, benang kotor, dan benang yang terinfeksi jamur. Kemudian gambar diambil dengan kamera webcam di bawah sinar uv lamp di dalam box hitam. Dengan posisi kamera tepat dipermukaan bagian atas gulungan benang seperti yang ditunjukkan pada gambar 2 di bawah ini.

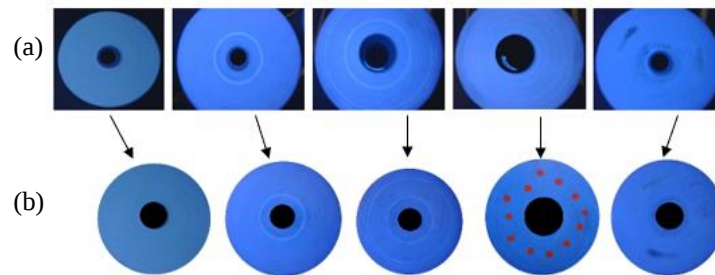


Gambar 2. Benang di Bawah UV Lamp.

Gambar yang sudah diambil tersebut (lihat Gambar 2) disimpan dalam format JPG untuk keperluan pra-pemrosesan citra selanjutnya.

2.2 Pra-Pemrosesan Citra

Pada tahap pra-pemrosesan citra ini, perangkat lunak seperti GNU Image Manipulation Program (GIMP) digunakan untuk meningkatkan kualitas citra yang diproses secara manual [10] seperti yang ditunjukkan pada Gambar 3. Langkah-langkah seperti penghapusan noise, penyesuaian kontras, pemotongan, dan pengubahan ukuran dilakukan untuk memastikan gambar sesuai dengan kebutuhan analisis pelatihan model machine learning nantinya.



Gambar 3. Pra-pemrosesan Citra: (a) citra asli yang ditangkap kamera; (b) citra dengan GIMP

Gambar 3(a) merupakan data image yang ditangkap secara langsung oleh camera dan disimpan dalam image classification database pada framework Teachable Machine [11], pada kondisi ini merupakan data riil dengan jumlah sample yang terbatas. Gambar 3(b) adalah image yang telah dimanipulasi atau dilakukan augmentasi untuk memperbanyak sample citra pada database. Setelah proses pra-pemrosesan selesai (lihat Gambar 3), gambar yang telah disesuaikan kemudian disimpan ke dalam folder masing-masing sesuai dengan nama atau label gambar. Langkah ini penting untuk memastikan data terorganisir dengan baik, sehingga dapat digunakan secara efektif dalam tahap pelatihan model machine learning karena gambar tersebut nantinya digunakan sebagai data training dan validasi.

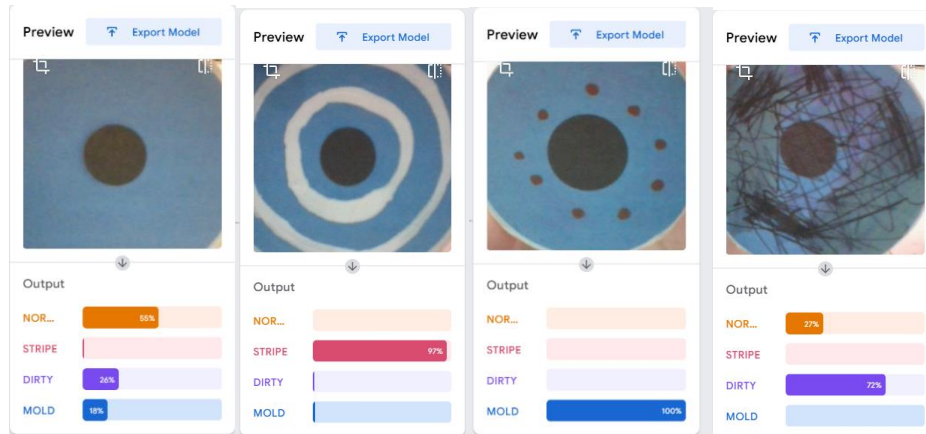
2.3 Pelatihan Model Machine Learning

Pelatihan model machine learning melibatkan penyesuaian bobot secara iteratif untuk meminimalkan kesalahan prediksi. Parameter utama seperti epoch (jumlah siklus pelatihan), batch size (jumlah sampel per iterasi), dan learning rate (besaran perubahan bobot) memengaruhi efisiensi dan akurasi model [12]. Epoch mengacu pada jumlah siklus penuh di mana model melihat seluruh dataset pelatihan; semakin banyak epoch dapat meningkatkan pembelajaran model, namun berisiko menyebabkan overfitting jika jumlahnya berlebihan. Batch size menentukan jumlah sampel yang diproses dalam satu iterasi, yang memengaruhi efisiensi memori serta kecepatan pelatihan. Sementara itu, learning rate mengatur besarnya perubahan bobot pada setiap iterasi, di mana nilai yang terlalu besar dapat menyebabkan model melewati solusi optimal [13]. Dalam Teachable Machine, pelatihan dimulai dengan mengunggah data yang dikategorikan ke dalam kelas seperti Normal, Stripe, Mold, dan Dirty. Model menyesuaikan bobotnya secara bertahap untuk meningkatkan akurasi, dengan hyperparameter yang dikonfigurasi untuk mencapai performa optimal sebelum evaluasi akhir.

2.4 Evaluasi Model Machine Learning

Evaluasi model machine learning bertujuan untuk menilai kemampuan model dalam menggeneralisasi pola dari data pelatihan menggunakan data yang belum pernah dilihat sebelumnya [14] [11]. Evaluasi dilakukan dengan data uji untuk memastikan model tidak

mengalami overfitting dan mampu membuat prediksi yang akurat. Pada Gambar 4, tahap preview dilakukan dengan data uji yang terdiri dari kategori benang normal, stripe, mold dan dirty. Tahap ini menjadi langkah penting dalam mengevaluasi kinerja model sebelum dideployment ke raspberry pi.



Gambar 4. Tahap Preview dengan Data Uji.

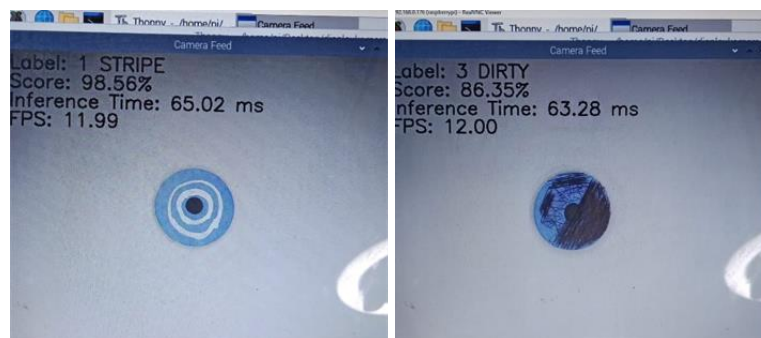
Setelah yakin dengan hasil preview seperti yang ditunjukkan oleh Gambar 4, model yang sudah dilatih dikonversi ke format TensorFlow Lite (TFLite).

2.5 Model Deployment ke Raspberry Pi

Model deployment ke Raspberry Pi merupakan tahap implementasi model yang telah dikonversi ke dalam format TensorFlow Lite ke perangkat Raspberry Pi [15]. Tahapan ini diawali dengan pemindahan model terlatih (.tflite) dan label ke Raspberry Pi, dilanjutkan dengan instalasi pustaka pendukung, seperti TensorFlow Lite Interpreter, guna memastikan kompatibilitas model dengan lingkungan eksekusi. Selanjutnya, dilakukan pengembangan skrip Python untuk memuat model, mengakses data dari kamera, serta menerapkan preprocessing citra sebelum dikirimkan ke model untuk proses inferensi. Hasil inferensi yang diperoleh kemudian dimanfaatkan untuk pengambilan keputusan, seperti penyajian informasi pada antarmuka pengguna.

2.6 Inferensi Model dengan Data Uji

Inferensi model pada Raspberry Pi digunakan untuk mengklasifikasikan data uji yang belum pernah dilatih, mengevaluasi kinerjanya dalam kondisi nyata. Model beroperasi real-time, memberikan respons cepat terhadap data dari kamera feed. Seperti pada Gambar 5, model mendeteksi objek dan memberi label "1 STRIPE" dengan kepercayaan 98.56%, serta menampilkan waktu inferensi 65.02 ms dan FPS 11.99 untuk mengukur performa pemrosesan citra.



Gambar 5. Inferensi Model dengan Data Uji Stripe dan Dirty

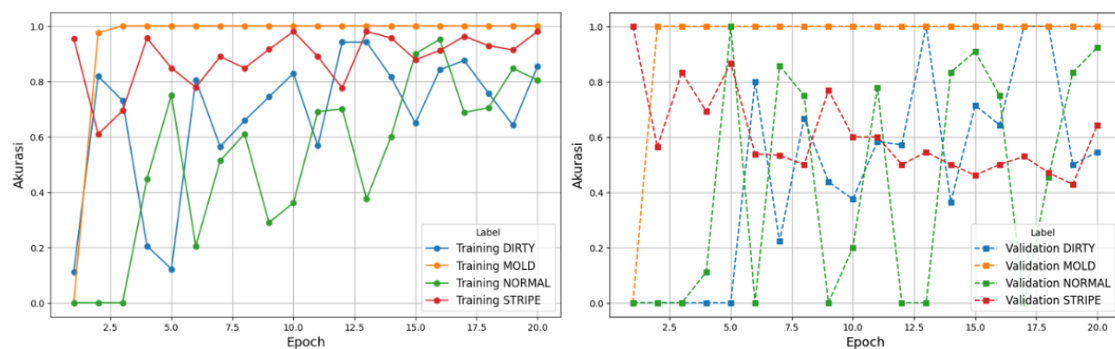
Data inferensi real-time yang ditampilkan pada Gambar 5 digunakan untuk menganalisis performa pemrosesan citra berdasarkan FPS, waktu inferensi, dan skor prediksi. Waktu inferensi menunjukkan durasi yang dibutuhkan model untuk menghasilkan prediksi, biasanya diukur dalam milidetik (ms). Semakin rendah waktu inferensi, semakin cepat model dalam memberikan hasil, meningkatkan efisiensi terutama dalam aplikasi real-time. Selain itu, Frames Per Second (FPS) mengindikasikan jumlah frame yang dapat diproses setiap detik. FPS yang tinggi memastikan sistem berjalan lebih lancar dan responsif, terutama dalam pemrosesan berbasis video atau aplikasi real-time. Skor prediksi, yang dinyatakan dalam probabilitas atau persentase, mencerminkan tingkat keyakinan model terhadap klasifikasi yang dilakukan [16]. Skor merupakan nilai yang mencerminkan tingkat keyakinan model terhadap prediksi atau klasifikasi yang dilakukan, biasanya dinyatakan dalam bentuk probabilitas atau persentase, yang membantu dalam mengevaluasi keakuratan dan keandalan model dalam melakukan tugas klasifikasi.

3. HASIL DAN PEMBAHASAN

Pada bab ini, akan dibahas mengenai hasil yang diperoleh dari pengujian yang telah dilakukan serta analisis terhadap performa model [17]. Hasil tersebut meliputi grafik akurasi dan loss yang menunjukkan perkembangan kinerja model pada data training dan validasi, serta tabel yang menggambarkan performa pemrosesan citra saat inferensi model dalam mengklasifikasikan jenis benang secara real-time. Pembahasan ini bertujuan untuk mengevaluasi efektivitas model dalam memproses dan mengklasifikasikan objek, serta untuk memahami faktor-faktor yang mempengaruhi hasil yang diperoleh.

3.1 Akurasi Model pada Data Training dan Validasi

Grafik di bawah menunjukkan perkembangan akurasi model pada data training dan validasi untuk setiap label benang. Sumbu X merepresentasikan jumlah epoch (iterasi pelatihan), sedangkan sumbu Y menunjukkan tingkat akurasi model seperti ditunjukkan pada Gambar 6.

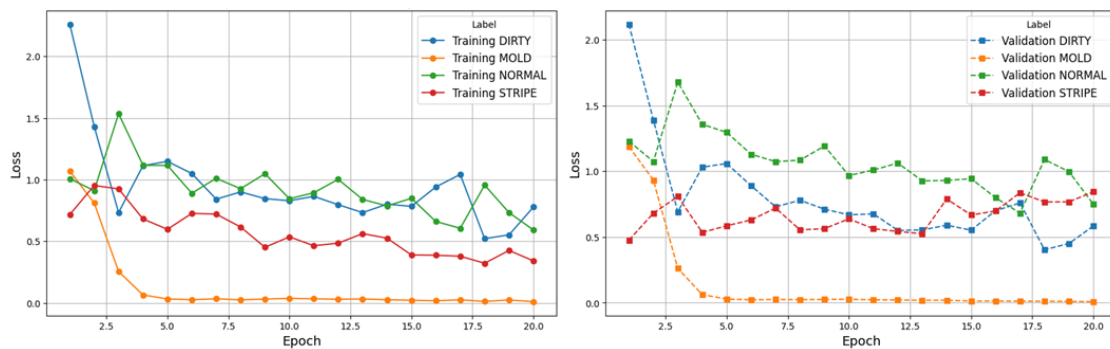


Gambar 6. Grafik Akurasi Model pada Data Training dan Validasi.

Pada Gambar 6, garis berwarna pada grafik merepresentasikan akurasi model pada data training dan validasi untuk masing-masing label, sementara titik-titik pada garis menunjukkan akurasi pada setiap epoch. Fluktuasi titik-titik tersebut mencerminkan perubahan akurasi dari satu epoch ke epoch berikutnya, yang dapat mengindikasikan pola belajar model serta stabilitas proses pelatihan.

3.2 Loss Model pada Data Training dan Validasi

Grafik yang ditunjukkan pada Gambar 7 menunjukkan perkembangan loss model pada data training dan validasi untuk setiap label benang. Sumbu X merepresentasikan jumlah epoch (iterasi pelatihan), sedangkan sumbu Y menunjukkan tingkat loss model seperti yang ditunjukkan pada Gambar 7.



Gambar 7. Grafik Loss Model pada Data Training dan Validasi.

Pada Gambar 7, garis berwarna pada grafik merepresentasikan loss model pada data training dan validasi untuk masing-masing label, sedangkan titik-titik pada garis menunjukkan nilai loss pada setiap epoch. Fluktuasi titik-titik tersebut mencerminkan perubahan loss dari satu epoch ke epoch berikutnya, yang dapat menggambarkan proses konvergensi model serta stabilitas pembelajaran selama pelatihan.

Akurasi merupakan perbandingan antara jumlah prediksi yang benar, baik untuk kelas positif maupun negatif, terhadap keseluruhan data yang digunakan dalam proses klasifikasi. Sementara itu, fungsi loss pada model berperan dalam mengevaluasi sejauh mana prediksi yang dihasilkan telah mendekati nilai sebenarnya, sehingga dapat digunakan sebagai indikator tingkat kesalahan dalam proses pembelajaran model [15]. Dari grafik akurasi dan loss di atas, didapat data akurasi dan loss model pada data training dan validasi yang disajikan dalam bentuk Tabel 1. Tabel 1 menunjukkan nilai loss dan akurasi model machine learning mulai dari yang paling rendah ke nilai yang paling tinggi pada data training dan validasi, yang memberikan gambaran lebih jelas tentang kinerja model dalam mengklasifikasikan objek pada setiap iterasi.

Tabel 1. Loss dan akurasi model pada data training dan validasi

No.	Label	Training Loss	Validation Loss	Training Accuracy	Validation Accuracy
1	Normal	0.6 - 1.6	0.7 - 1.7	0.1 - 0.9	0.0 - 1.0
2	Stripe	0.4 - 0.8	0.5 - 0.8	0.7 - 1.0	0.5 - 1.0
3	Mold	0.0 - 1.1	0.0 - 1.2	0.0 - 1.0	0.0 - 1.0
4	Dirty	0.5 - 2.2	0.4 - 2.2	0.1 - 0.9	0.0 - 1.0

Tabel 1 menyajikan rentang nilai training loss, validation loss, training accuracy, dan validation accuracy untuk masing-masing label (Normal, Stripe, Mold, dan Dirty), yang digunakan untuk menganalisis kinerja model dalam mengklasifikasikan data. Untuk label "Normal", meskipun training loss mengalami penurunan, validation loss tetap tinggi, yang menunjukkan potensi overfitting. Sementara itu, training accuracy meningkat, namun validation accuracy bervariasi, menandakan ketidakstabilan dalam prediksi. Label "Stripe" menunjukkan training loss yang rendah dan validation loss yang stabil, dengan akurasi yang meningkat hingga 1.0, mengindikasikan kemampuan model yang baik dalam mengenali label ini meskipun masih ada kemungkinan kesalahan. Pada label "Mold," training dan validation loss yang mencapai nilai rendah menandakan model dapat mengenali pola dengan baik, namun adanya potensi overfitting tercermin dari akurasi yang mencapai 1.0. Untuk label "Dirty," meskipun training loss turun dari nilai yang sangat tinggi, validation loss tetap tinggi, menunjukkan kesulitan dalam generalisasi, sementara akurasi yang fluktuatif menunjukkan ketidakstabilan dalam mengenali kategori ini.

3.3 Pengujian Performa Pemrosesan Citra Real-Time

Pengujian ini dilakukan untuk mengevaluasi performa model dalam melakukan inferensi secara real-time berbagai label benang dengan data uji. Parameter yang diukur mencakup FPS (Frames Per Second), waktu inferensi (inference time), dan skor prediksi untuk setiap label. Hasil pengujian disajikan dalam Tabel 2, yang memberikan gambaran mengenai efisiensi dan stabilitas pemrosesan citra pada masing-masing kategori label.

Tabel 2. Perbandingan score, FPS, dan inference time untuk setiap label

Label	Inference Time (ms)	Score (%)	FPS
Normal	55 - 75	52 - 66	11.15 - 11.5
Stripe	50 - 85	90 - 100	11.1 - 11.6
Mold	50 - 80	95 - 100	2 - 10.5
Dirty	50 - 100	65 – 95	2 - 10.5

Tabel 2, menunjukkan nilai rentang skor, FPS (Frames Per Second), dan waktu inferensi (inference time) untuk masing-masing label, yang memberikan gambaran tentang kinerja model dalam memproses data. Untuk label "Normal," skor akurasi prediksi model cukup rendah dan fluktuatif, sementara FPS stabil namun inferensi time bervariasi, menunjukkan ketidakstabilan dalam pemrosesan. Label "Stripe" menunjukkan skor yang konsisten tinggi, dengan FPS yang stabil dan tinggi, serta waktu inferensi yang cukup stabil dengan sedikit fluktuasi, mencerminkan efisiensi pemrosesan. Pada label "Mold," akurasi sangat tinggi dengan stabilitas luar biasa, namun FPS rendah di awal yang kemudian meningkat, dan waktu inferensi yang stabil setelah fluktuasi awal, menunjukkan efisiensi pemrosesan yang meningkat setelah stabilisasi. Label "Dirty" memiliki skor akurasi yang lebih bervariasi namun cenderung tinggi, dengan FPS yang rendah di awal namun meningkat seiring waktu, serta waktu inferensi yang fluktuatif namun stabil setelah beberapa frame.

4. KESIMPULAN

Hasil inferensi model pada sistem, performa klasifikasi menunjukkan perbedaan signifikan antar kategori. Model CNN menunjukkan akurasi tinggi dalam mengenali kategori Mold (95-100%) dan Stripe (90-100%), namun kurang konsisten dalam membedakan Normal (52-66%) dan Dirty (65-95%). Dari segi kecepatan pemrosesan (FPS), kategori Stripe paling stabil (11.1-11.6 FPS), sementara Dirty dan Mold mengalami penurunan FPS di awal inferensi sebelum stabil. Analisis akurasi dan loss menunjukkan potensi overfitting pada Mold, sementara kategori Normal dan Dirty memiliki validation loss tinggi dan akurasi yang tidak stabil. Stripe menunjukkan performa lebih baik dengan validation loss yang lebih stabil. Untuk meningkatkan generalisasi model, perlu dilakukan penyesuaian arsitektur CNN, optimalisasi preprocessing gambar, peningkatan teknik data augmentation, serta fine-tuning hyperparameter guna meningkatkan akurasi pada kategori yang masih lemah.

DAFTAR PUSTAKA

- [1] A. Yulianto and E. Ramadan, "Sistem Kendali Robot Manipulator Pemindah Barang Dengan Umpan Balik Visual," *J. Ilm. Mikrotek*, vol. 1, no. 2, pp. 1–8, 2014, [Online]. Available: <http://kompetensi.trunojoyo.ac.id/jim/article/viewFile/157/154>
- [2] M. Convolutional and N. Networks, "Indonesian Journal of Education Penerapan Deep Learning untuk Klasifikasi Buah Berdasarkan," vol. 2, no. 3, pp. 196–202, 2024.

-
- [3] Tri Wahyu Qur'ana, "Implementasi Metode Convolutional Neural Network (CNN) untuk Klasifikasi Motif pada Citra Sasirangan," *Media Inform.*, vol. 7, no. 2, p. 10, 2023, [Online]. Available: <http://jurnal.big.go.id/index.php/GM/article/view/810>
- [4] R. K. Dianti, R. Regasari, M. Putri, and E. Setiawan, "Klasifikasi Kanker Kulit Jinak Dan Ganas Menggunakan Metode Xception Berbasis Raspberry Pi," vol. 9, no. 4, pp. 1–11, 2025.
- [5] F. W. Prabowo *et al.*, "DETEKSI WARNA KULIT MENGGUNAKAN METODE DEEP LEARNING DENGAN CNN (CONVOLUTIONAL NEURAL NETWORK) UNTUK," vol. 19, no. September, pp. 186–190, 2024.
- [6] B. Yanto, E. Rouza, L. Fimawahib, B. H. Hayadi, and R. R. Pratama, "Penerapan Algoritma Deep Learning Convolutional Neural Network Dalam Menentukan Kematangan Buah Jeruk Manis Berdasarkan Citra Red Green Blue (RGB)," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 10, no. 1, pp. 59–66, 2023, doi: 10.25126/jtiik.20231015695.
- [7] H. Darmanto, "Pengenalan Spesies Ikan Berdasarkan Kontur Otolith Menggunakan Convolutional Neural Network," *Joined J. (Journal Informatics Educ.*, vol. 2, no. 1, p. 41, 2019, doi: 10.31331/joined.v2i1.847.
- [8] A. R. Nugraha, F. Utaminigrum, and H. Fitriyah, "Sistem Deteksi Hama Babi menggunakan CNN (Convolutional Neural Network) Berbasis Raspberry Pi," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 9, pp. 3645–3651, 2021, [Online]. Available: <http://j-ptiik.ub.ac.id>
- [9] M. K. Mawaddah Harahapp and M. K. Amir Mahmud Husein, "PENERAPAN EFFICIENT-NET DALAM MENGLASIFIKASI KANJER KULIT," *Monograft*, pp. 1–66, 2024.
- [10] N. Puspitasari, A. Septiarini, and A. R. Aliudin, "Metode K-Nearest Neighbor Dan Fitur Warna Untuk Klasifikasi Daun Sirih Berdasarkan Citra Digital," *PROSISKO J. Pengemb. Ris. dan Obs. Sist. Komput.*, vol. 10, no. 2, pp. 165–172, 2023, doi: 10.30656/prosisko.v10i2.6924.
- [11] E. A. Nugroho, J. D. Setiawan, Munadi, and Diki, "Design of image classification system for fabric inspection process using Raspberry Pi," *J. Mechatronics, Electr. Power, Veh. Technol.*, vol. 15, no. 1, pp. 57–67, 2024, doi: 10.55981/j.mev.2024.863.
- [12] L. Satya, M. R. D. Septian, M. W. Sarjono, M. Cahyanti, and E. R. Swedia, "Sistem Pendeteksi Plat Nomor Polisi Kendaraan Dengan Arsitektur Yolov8," *Sebatik*, vol. 27, no. 2, pp. 753–761, 2023, doi: 10.46984/sebatik.v27i2.2374.
- [13] Afis Julianto, Andi Sunyoto, and Ferry Wahyu Wibowo, "Optimasi Hyperparameter Convolutional Neural Network Untuk Klasifikasi Penyakit Tanaman Padi," *Tek. Teknol. Inf. dan Multimed.*, vol. 3, no. 2, pp. 98–105, 2022, doi: 10.46764/teknimedia.v3i2.77.
- [14] E. Setia Budi, A. Nofriyaldi Chan, P. Priscillia Alda, and M. Arif Fauzi Idris, "RESOLUSI : Rekayasa Teknik Informatika dan Informasi Optimasi Model Machine Learning untuk Klasifikasi dan Prediksi Citra Menggunakan Algoritma Convolutional Neural Network," *Media Online*, vol. 4, no. 5, p. 509, 2024, [Online]. Available: <https://djournal.com/resolusi>
- [15] E. A. Nugroho *et al.*, "Penerapan Teachable Machine dan Raspberry Pi pada Sistem Klasifikasi Citra untuk Inspeksi Cacat Kain.," *JTIK*, vol. 12, no. 1, pp. 77–86, 2025, doi: 10.25126/jtiik.2025128932.
- [16] C. Mwitata, G. C. Rains, and E. Prostko, "Evaluation of Inference Performance of Deep Learning Models for Real-Time Weed Detection in an Embedded Computer," *Sensors*, vol. 24, no. 2, 2024, doi: 10.3390/s24020514.
- [17] F. Zaelani and Y. Miftahuddin, "Perbandingan Metode EfficientNetB3 dan MobileNetV2 Untuk Identifikasi Jenis Buah-buahan Menggunakan Fitur Daun," *J. Ilm. Teknol. Infomasi Terap.*, vol. 9, no. 1, pp. 1–11, 2022, doi: 10.33197/jitter.vol9.iss1.2022.911.