

DETEKSI HELM SAFETY MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORKS DAN YOLOV3-TINY

Dicky Arya Aji Prayoga¹, Muhammad Taufiqurrochman Abdul Aziz Zein², M. Khanif³

Program Studi Informatika, Universitas Nahdlatul Ulama Al Ghazali Cilacap
Jl. Kemerdekaan Barat No 17 RT.01 Rw.05 Kesugihan Kidul, Kab Cilacap, Jawa Tengah

¹dickyaryaap@gmail.com

²zein@unugha.id

³khanif.clp123@gmail.com

Dikirim pada 17-10-2024, Direvisi pada 28-10-2024, Diterima pada 10-11-2024

Abstrak

Dalam dunia industri keselamatan seseorang menjadi perhatian utama, terutama terkait penggunaan helm *safety* sebagai alat pelindung diri. Untuk meningkatkan pengawasan penggunaan helm *safety* secara otomatis, teknologi digital image processing menawarkan solusi inovatif. Penggunaan metode *deep learning*, khususnya dengan algoritma *Convolutional Neural Network* (CNN), mampu menganalisis citra secara efektif dan mendeteksi keberadaan objek dengan akurasi tinggi. Salah satu model deteksi yang digunakan dalam penelitian ini adalah You Only Look Once (YOLO) yang dirancang untuk kecepatan dan efisiensi. Dengan menggabungkan kemampuan deteksi objek YOLOv3 Tiny dan arsitektur CNN yang kuat, sistem dapat memonitor dan mengidentifikasi penggunaan helm *safety* secara *real-time*, bahkan dalam lingkungan kerja yang dinamis dan kompleks. Penelitian ini menggunakan beberapa tahap utama, yakni pengumpulan data, pra-pemrosesan, pelatihan, dan pengujian. Pada tahap awal, gambar diambil dari berbagai sudut untuk mendokumentasikan objek yang mengenakan helm *safety* dan yang tidak. Selanjutnya, dalam proses prapemrosesan, objek-objek dalam gambar diberi label menggunakan aplikasi LabelImg. Total terdapat 708 anotasi untuk gambar objek yang mengenakan helm *safety*, dan 794 untuk yang tidak mengenakan helm *safety*. Pelatihan model dilakukan menggunakan metode *transfer learning* dengan arsitektur *darknet-53* YOLOv3-tiny, yang dijalankan pada platform *Google Collaboratory*. Untuk mendeteksi objek, *Convolutional Neural Network* (CNN) digunakan pada tahap pengenalan objek, sedangkan YOLOv3-tiny diterapkan pada proses deteksi. Pengujian model dilakukan dengan teknik *10-fold cross-validation*, dengan parameter pelatihan *epoch* sebanyak 50 dan batch size 32. Hasil pengujian menunjukkan bahwa model mampu mencapai akurasi sebesar 0,8, dengan nilai presisi 0,83, recall 0,8, dan F1 score 0,8.

Kata Kunci: Helm *Safety*, CNN, Yolov3 Tiny, Deteksi.

Ini adalah artikel akses terbuka di bawah lisensi [CC BY-SA](#).



Penulis Koresponden:

Dicky Arya Aji Prayoga

Program Studi Informatika, Universitas Nahdlatul Ulama Al Ghazali Cilacap. Jl. Kemerdekaan Barat No.17, Gligir, Kesugihan Kidul, Kec. Kesugihan, Kabupaten Cilacap, Jawa Tengah 53274 Email: dickyaryaap@gmail.com

I. PENDAHULUAN

Seiring dengan pesatnya perkembangan teknologi, pengolahan citra digital telah mengalami kemajuan yang signifikan di berbagai bidang. Salah satu teknik yang memainkan peran penting dalam kemajuan ini adalah Digital Image Processing (DIP) [1], yaitu metode untuk menangkap dan mengolah gambar tanpa merusak objek secara langsung. Digital Image Processing (DIP) melibatkan penggunaan perangkat keras untuk menangkap citra, yang kemudian diolah dengan perangkat lunak menggunakan algoritma tertentu. Teknologi ini memungkinkan pengelolaan citra digital secara efisien, termasuk dalam proses klasifikasi objek [2]. Citra digital memiliki peran penting dalam dunia multimedia, terutama karena kemampuannya

dalam menyampaikan informasi secara visual. Penggunaan gambar dalam menyampaikan informasi sering kali lebih efektif dibandingkan hanya teks. Dalam pengolahan citra digital, metode ini biasanya diterapkan untuk meningkatkan kualitas gambar agar lebih mudah diinterpretasikan, baik oleh manusia maupun komputer. Salah satu metode paling populer dalam pengolahan citra digital adalah Convolutional Neural Network (CNN) [3]. Convolutional Neural Network (CNN) merupakan salah satu algoritma dari Deep Learning yang merupakan pengembangan dari Multi Layer Perceptron (MLP) [4]. Algoritma ini telah terbukti sangat efektif dalam pengenalan gambar dan klasifikasi objek, seperti identifikasi benda [5], pengenalan pola [6], dan pengenalan wajah [7]. Berbagai penelitian menunjukkan bahwa CNN memberikan hasil akurasi yang sangat tinggi. Salah satunya penelitian tentang klasifikasi tekstur kematangan buah jeruk manis yang menggunakan metode Deep Learning berbasis citra digital menunjukkan akurasi sebesar 96% pada data pelatihan dan 92% pada pengujian. Penelitian lain juga menemukan bahwa CNN sangat efektif dalam klasifikasi tanaman dengan menggunakan citra resolusi tinggi, mencapai akurasi sebesar 93% pada data validasi [8]. Penelitian lain juga mengeksplorasi klasifikasi tanaman menggunakan citra resolusi tinggi. Hasil pengujian terhadap data validasi mencapai akurasi 93%, sementara akurasi pada data tes adalah 82%. Temuan ini menunjukkan bahwa metode Convolutional Neural Network (CNN) memiliki potensi besar dalam pendekatan pengenalan objek secara otomatis, terutama dalam membedakan jenis tanaman, yang dapat menjadi bahan pertimbangan penting bagi interpreter dalam menentukan objek pada citra [9].

Salah satu arsitektur CNN yang banyak digunakan dalam deteksi objek adalah YOLO (You Only Look Once), yang dikenal karena kecepatan dan akurasinya dalam mendeteksi objek. Meskipun YOLOv3-Tiny memiliki akurasi yang sedikit lebih rendah dibandingkan versi YOLOv3 dan YOLOv4 [10], arsitektur ini menawarkan kecepatan pemrosesan yang lebih tinggi dengan biaya komputasi yang lebih rendah. Karena itu, YOLOv3-Tiny dinilai lebih efisien untuk aplikasi yang membutuhkan deteksi real-time seperti monitoring penggunaan helm safety.

Dalam konteks deteksi helm safety menggunakan computer vision [11], teknologi ini memungkinkan identifikasi pekerja konstruksi yang tidak menggunakan helm safety dengan tingkat akurasi dan efisiensi yang tinggi. Dibandingkan dengan metode manual, sistem berbasis YOLOv3-Tiny menawarkan solusi yang lebih cepat dan dapat diandalkan dalam situasi yang dinamis. Penelitian ini menggunakan metode pengembangan dan evaluasi model deteksi objek berbasis Deep Learning, dengan tahap pelatihan dilakukan menggunakan metode 10-Fold Cross Validation untuk mengevaluasi kinerja model yang dirancang [12].

A. Pendahuluan

Dalam sektor industri terkini, khususnya di bidang konstruksi dan manufaktur. Salah satu alat pelindung diri (APD) yang wajib digunakan oleh pekerja adalah helm keselamatan. Helm ini bertujuan untuk melindungi kepala dari risiko cedera yang disebabkan oleh kecelakaan atau benda jatuh. Walaupun aturan ketat terkait penggunaan helm keselamatan sudah diberlakukan, masih banyak pekerja yang kurang disiplin dalam memakainya. Pengawasan manual oleh petugas keselamatan juga sering kali tidak efektif dan memerlukan banyak sumber daya. Tantangan utama dalam pengawasan pemakaian helm keselamatan adalah ketergantungan pada metode manual yang memakan waktu, tenaga, serta biaya yang tidak sedikit. Pelanggaran terhadap aturan ini dapat menyebabkan cedera parah, bahkan kematian. Oleh karena itu, dibutuhkan solusi yang lebih efektif dan efisien untuk memastikan bahwa pekerja selalu memakai helm keselamatan di area kerja.

Salah satu pendekatan yang diajukan adalah pengembangan sistem pengawasan otomatis yang memanfaatkan teknologi visi komputer, khususnya Convolutional Neural Networks (CNN) dan algoritma YOLOv3-Tiny [13]. CNN telah terbukti handal dalam mengenali objek melalui citra, sementara YOLOv3-Tiny, yang merupakan varian ringan dari YOLO, mampu mendeteksi objek secara real-time dengan kinerja yang cepat dan penggunaan sumber daya yang lebih hemat. Dengan memadukan kedua teknologi ini, sistem pengawasan otomatis dapat mendeteksi secara langsung apakah pekerja menggunakan helm keselamatan atau tidak, sehingga proses pengawasan menjadi lebih efisien.

Sistem ini diharapkan dapat meningkatkan kepatuhan pekerja dalam menggunakan helm keselamatan, sekaligus mengurangi risiko kecelakaan akibat kelalaian. Selain itu, dari segi operasional, sistem ini memberikan keuntungan dengan mampu melakukan pemantauan lapangan secara terus-menerus tanpa memerlukan pengawasan manusia yang intensif.

II. METODE PENELITIAN

Penelitian ini diawali dengan pada tahap pengumpulan data penggunaan helm safety pada gambar dan video menggunakan salah satu metode *deep learning* yaitu yolov3. Dataset untuk data latih pada penelitian ini berupa gambar orang yang memakai helm safety dan yang tidak. Jumlah data latih sebanyak 448 gambar yang terdiri dari 209 gambar orang yang memakai helm dan 239 gambar orang yang tidak memakai helm.

A. Data Collection

Dalam penelitian ini, dataset yang digunakan merupakan dataset sekunder, yang diperoleh dari platform maedeley dataset [14], yaitu *Helmet Detection Dataset*. Dataset ini terdiri dari gambar-gambar mengenakan helm *safety* dan yang tidak, yang diambil dalam berbagai kondisi pencahayaan dan sudut pandang. Dataset ini dipilih karena telah dilabeli secara manual dan telah digunakan dalam berbagai penelitian terkait deteksi objek dengan *computer vision*, sehingga menyediakan data yang cukup untuk melatih dan menguji model deteksi helm. Gambar-gambar tersebut diambil dengan variasi kondisi lingkungan, seperti pencahayaan yang berbeda (dari siang hari hingga malam hari), serta sudut pandang kamera yang bervariasi (0 hingga 90 derajat). Keberagaman kondisi dalam dataset ini memberikan tantangan tambahan dalam proses pelatihan model, yang diharapkan dapat meningkatkan akurasi dan generalisasi model deteksi objek. Pada tahap pra-pemrosesan, gambar-gambar dalam dataset ini telah dibagi ke dalam dua kelas: helm *safety* dan tanpa helm. Setiap gambar telah dilabeli dengan bounding box untuk menandai keberadaan objek helm di dalam gambar, yang mempermudah model dalam belajar untuk mengenali objek tersebut dalam berbagai kondisi.

B. Flowchart Alur Deteksi

Pada alur sistem yang digunakan dalam penelitian ini ditunjukkan dalam *flowchart* yang digambarkan pada Gambar 1. Proses awal adalah mengumpulkan data, memberi label pada gambar, lalu melakukan pelatihan dengan *transfer learning* menggunakan model pra-latihan YOLOv3-Tiny untuk menghasilkan bobot baru. Selanjutnya, bobot yang tidak diketahui digunakan untuk mendeteksi objek yang mengenakan helm dan yang tidak muncul dalam gambar percobaan.



Gambar 1. Alur Deteksi

Proses pelatihan model dilakukan dengan menggunakan arsitektur darknet-53 sebagai kerangka kerja dan YOLOv3-tiny sebagai bobot awal. Teknik transfer learning diterapkan untuk memperbarui bobot model, yaitu dengan memanfaatkan bobot model yang telah dilatih sebelumnya agar dapat mengenali objek baru [15]. Dalam hal ini, bobot model YOLOv3-tiny yang digunakan berjalan pada platform Google Colaboratory, dengan pengaturan tertentu untuk epoch dan batch size.

Langkah pertama dalam proses ini adalah menghubungkan Google Colab pada Google Drive. Setelah itu, berkas konfigurasi darknet diunduh dari GitHub, kemudian GPU, OpenCV, dan cuDNN diaktifkan untuk mempercepat pemrosesan. Selanjutnya, darknet dikonfigurasi dengan YOLOv3-tiny dan gambar-gambar diekstrak dari file .zip untuk memulai proses pelatihan model. Pelatihan ini berlangsung selama kurang lebih 3 jam, dan hasilnya berupa bobot model yang diperbarui disimpan di Google Drive.

C. Flowchart Alur Deteksi

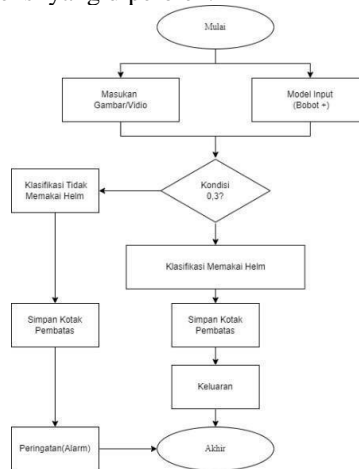
Dataset yang telah diberi anotasi kemudian digunakan dalam proses pelatihan model. Pada tahap pelatihan ini gambar 3, Jaringan Syaraf Tiruan (JST) dilatih untuk mengenali pola-pola yang terdapat dalam gambar beserta anotasinya. Dengan pola-pola tersebut, JST akan belajar melakukan pengenalan objek yang relevan, yang nantinya akan digunakan dalam pengembangan sistem deteksi objek secara otomatis.



Gambar 2. Training

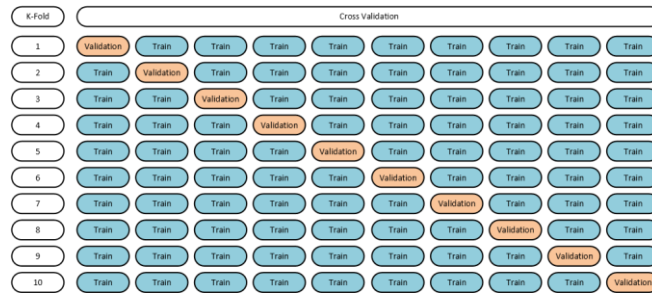
D. Deteksi

Pada tahap deteksi, YOLO memproses gambar input dengan ukuran jaringan yang merupakan kelipatan dari 32 [16]. Semakin besar ukuran jaringan yang digunakan, semakin tinggi akurasi deteksi yang bisa diprediksi oleh komputer. Namun, peningkatan ukuran jaringan ini akan memperlambat proses komputasi. Sebaliknya, jika ukuran jaringan diperkecil, proses komputasi akan menjadi lebih cepat, tetapi dengan risiko penurunan akurasi deteksi yang diperoleh.



Gambar 3. Deteksi

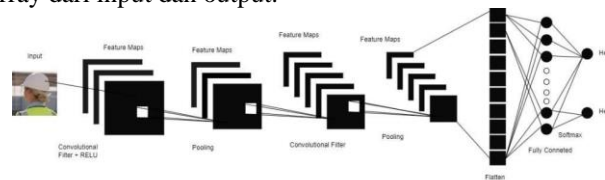
Dalam penelitian ini, ukuran jaringan yang digunakan untuk deteksi adalah 416×416 piksel. Selanjutnya dilakukan proses ekstraksi fitur menggunakan arsitektur darknet-53, yang terdiri dari 53 lapisan konvolusional. Lapisan konvolusional ini adalah komponen utama dalam struktur Convolutional Neural Network (CNN) dan merupakan tempat sebagian besar komputasi dilakukan. Pada proses ekstraksi fitur, setiap lapisan konvolusional di darknet-53 menggunakan fungsi aktivasi Leaky ReLU, kecuali lapisan terakhir yang menggunakan fungsi aktivasi linear. YOLOv3 menghasilkan prediksi dalam bentuk enam nilai koordinat bounding box (tx, ty, tw, th), keyakinan probabilitas, dan kelas objek. Untuk memprediksi kelas dari bounding box, YOLOv3 menggunakan klasifikasi multilabel. Sistem ini menghitung skor kelas setiap bounding box menggunakan regresi logistik, yang membantu mencegah adanya tumpang tindih antara bounding box. Jika skor kelas dari bounding box lebih tinggi dari nilai keyakinan yang telah ditentukan, maka objek tersebut diberikan kelas sesuai hasil deteksi. Diagram alur proses deteksi dapat dilihat pada Gambar 3.



Gambar 4. Metode k-fold cross-validation

E. Arsitektur *Convolutional Neural Network*

Convolutional Neural Network (CNN) adalah salah satu metode deep learning yang digunakan untuk mendeteksi, mengidentifikasi, dan mengklasifikasikan data gambar. CNN sangat populer dalam pemrosesan citra karena mampu mencapai tingkat akurasi yang tinggi. Model ini terbagi menjadi dua komponen utama, yaitu proses ekstraksi fitur dan klasifikasi. CNN juga memiliki arsitektur yang terdiri dari beberapa tahapan yang dapat dilatih secara berurutan. Pada setiap tahapan, terdapat peta fitur yang merupakan kumpulan array dari input dan output.



Gambar 5. Arsitektur Jaringan

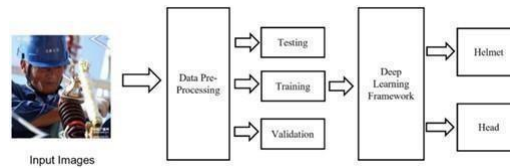
CNN memiliki tiga lapisan utama di setiap tahapannya. Pertama, lapisan konvolusi, di mana kernel dengan ukuran tertentu digunakan untuk menghasilkan sejumlah fitur. Tahap kedua melibatkan fungsi aktivasi, di mana unit linear rectifier atau ReLU digunakan pada tahap terakhir adalah pooling, di mana informasi yang dihasilkan dari proses konvolusi diringkas untuk mengurangi dimensi data. Hasil dari proses ini biasanya disebut sebagai keluaran atau representasi fitur.

F. Deteksi

Pada tahap ini, dijelaskan arsitektur model deteksi helm keselamatan menggunakan metode Deep Learning. Arsitektur sistem ini terdiri dari beberapa langkah penting, dimulai dari input gambar hingga menghasilkan keluaran berupa deteksi objek helm keselamatan. Tahap pertama dalam sistem ini adalah menerima input gambar, yang bisa berasal dari foto, video, atau kamera pengawas. Gambar ini menjadi dasar untuk menentukan apakah objek dalam gambar mengenakan helm keselamatan atau tidak. Setelah gambar diterima, gambar tersebut akan melalui tahap prapemrosesan data. Tujuan dari tahap ini adalah mempersiapkan gambar agar sesuai dengan format yang dibutuhkan oleh model deteksi objek.

Setelah data diproses, langkah selanjutnya adalah membagi data tersebut menjadi tiga bagian utama, yaitu *training set*, *validation set*, dan *testing set*. Pembagian ini penting untuk memastikan model dilatih dengan baik, divalidasi secara tepat, dan diuji untuk mengukur performanya. Setelah melalui proses pembagian data, gambar diproses menggunakan framework *Deep Learning*. Dalam hal ini, YOLOv3-Tiny digunakan sebagai arsitektur deteksi objek. YOLOv3-Tiny merupakan versi yang lebih ringan dari YOLOv3 [17], dirancang khusus untuk deteksi *real-time* dengan latensi rendah. Model ini sangat cocok untuk aplikasi yang membutuhkan kecepatan dan efisiensi.

Hasil akhir dari proses ini adalah deteksi helm dan deteksi kepala. Sistem akan memberikan keluaran berupa *bounding box* yang mengelilingi helm jika terdeteksi pada gambar, serta label yang menyatakan objek tersebut sebagai "helm." Jika helm tidak terdeteksi, sistem akan mendeteksi kepala dan mengeluarkan *bounding box* dengan label "kepala." Keluaran sistem ini dapat digunakan untuk pengawasan otomatis secara *real-time*.



Gambar 6. Hasil Deteksi

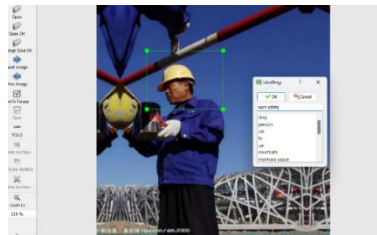
III. HASIL DAN PEMBAHASAN

Pada bab ini, menjejelaskan arsitektur model deteksi helm *safety*, kecepatan deteksi dan akurasi dengan objek *real-time* menggunakan *metode Deep Learning*.

A. Deteksi

Pada tahap ini, dijelaskan arsitektur model deteksi helm keselamatan menggunakan metode *Deep Learning*. Arsitektur sistem ini terdiri dari beberapa langkah penting, dimulai dari input gambar hingga menghasilkan keluaran berupa deteksi objek helm keselamatan. Tahap pertama dalam sistem ini adalah menerima input gambar, yang bisa berasal dari foto, video, atau kamera pengawas. Gambar ini menjadi dasar untuk menentukan apakah objek dalam gambar mengenakan helm keselamatan atau tidak. Setelah gambar diterima, gambar tersebut akan melalui tahap prapemrosesan data. Tujuan dari tahap ini adalah mempersiapkan gambar agar sesuai dengan format yang dibutuhkan oleh model deteksi objek.

Setelah dataset terkumpul dan dirapikan dalam folder sesuai kelas, serta melalui tahap prapemrosesan, langkah selanjutnya adalah membagi dataset yang telah ada menjadi tiga kategori, yaitu latih, uji, dan validasi. Dataset kemudian diberi label sesuai dengan kelas/folder/kategori. Selanjutnya, dataset diuji menggunakan model yang sudah tersedia pada library Python.



Gambar 7. Pelabelan Dataset Helm Safety

Pada tahap ini, angka 0 0.488021 0.276852 0.130208 0.316667 merepresentasikan sebuah objek dengan ID kelas 0 (kemungkinan seorang manusia). Objek ini memiliki titik pusat pada koordinat (0.488021, 0.276852) dalam skala normalisasi. Lebar bounding box yang mengelilingi objek tersebut adalah 0.130208 dari keseluruhan lebar gambar, sedangkan tingginya adalah 0.316667 dari total tinggi gambar. Format ini adalah standar yang digunakan oleh YOLO untuk melatih model deteksi objek berdasarkan data yang telah dilabeli.

B. Model CNN

Setelah tahap pelabelan dan pengambilan data pelatihan dari dataset, model akan dilatih menggunakan algoritma Convolutional Neural Network (CNN). Dalam CNN ini, digunakan model Sequential seperti yang ditunjukkan pada gambar, yang merupakan arsitektur jaringan saraf konvolusional yang digunakan untuk tugas klasifikasi dengan kemungkinan dua kelas keluaran. Model ini terdiri dari beberapa lapisan utama yang bekerja sama untuk mengekstrak fitur dari gambar dan melakukan klasifikasi akhir.

Model ini dimulai dengan tiga lapisan konvolusi (Conv2D), masing-masing bertujuan untuk mengekstrak fitur dari input gambar melalui filter konvolusi. Setiap lapisan konvolusi diikuti oleh lapisan MaxPooling2D yang berfungsi untuk mengurangi dimensi spasial dari peta fitur yang dihasilkan. Hal ini

tidak hanya mengurangi ukuran data yang diproses, tetapi juga menurunkan kompleksitas komputasi, sehingga membuat proses pelatihan menjadi lebih efisien.

Lapisan pertama menggunakan 16 filter, diikuti oleh 32 filter pada lapisan kedua, dan 64 filter pada lapisan ketiga. Semakin banyak filter yang digunakan, semakin kompleks pola yang dapat dipelajari oleh model dari gambar input. Setelah proses konvolusi dan pooling selesai, output dari lapisan terakhir memiliki dimensi (6x6x64), yang kemudian diratakan menggunakan lapisan Flatten untuk mengubahnya menjadi vektor satu dimensi. Setelah lapisan Flatten, model memiliki tiga lapisan dense (fully connected). Lapisan dense pertama terdiri dari 200 neuron, diikuti oleh lapisan dropout untuk mencegah overfitting dengan menonaktifkan beberapa neuron secara acak selama pelatihan. Kemudian, terdapat lapisan dense kedua dengan 500 neuron, diikuti lagi oleh lapisan dropout.

Lapisan terakhir dari model ini adalah lapisan dense dengan 2 neuron, yang berperan sebagai keluaran untuk tugas klasifikasi biner. Kedua neuron ini mewakili dua kelas yang akan diprediksi oleh model. Model ini memiliki total 586.086 parameter, semuanya adalah parameter yang dapat dilatih. Tidak ada parameter tetap atau non-trainable. Jumlah parameter yang besar ini menunjukkan kapasitas model untuk mempelajari pola yang kompleks dari data pelatihan. Secara keseluruhan, model ini merupakan implementasi jaringan saraf konvolusional yang cukup sederhana namun efektif untuk tugas klasifikasi gambar. Kombinasi lapisan konvolusi dan pooling memungkinkan model untuk mengekstrak fitur relevan dari gambar, sementara lapisan dense digunakan untuk melakukan prediksi akhir. Adanya lapisan dropout membantu mengurangi risiko overfitting, sehingga model diharapkan dapat menghasilkan prediksi yang lebih baik pada data baru yang belum pernah dilihat sebelumnya.

```
model.summary()
Model: "sequential"
-----
```

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 62, 62, 16)	448
max_pooling2d (MaxPooling2D)	(None, 31, 31, 16)	0
conv2d_1 (Conv2D)	(None, 29, 29, 32)	4640
max_pooling2d_1 (MaxPooling2D)	(None, 14, 14, 32)	0
conv2d_2 (Conv2D)	(None, 12, 12, 64)	18496
max_pooling2d_2 (MaxPooling2D)	(None, 6, 6, 64)	0
flatten (Flatten)	(None, 2384)	0
dense (Dense)	(None, 200)	461000
dropout (Dropout)	(None, 200)	0
dense_1 (Dense)	(None, 500)	100500
dropout_1 (Dropout)	(None, 500)	0
dense_2 (Dense)	(None, 2)	1002

```
-----
Total params: 586,086
Trainable params: 586,086
Non-trainable params: 0
```

Gambar 8. Model CNN

C. Hasil Akurasi

Pada tahap ini dilakukan uji akurasi

```
model.compile(loss = 'categorical_crossentropy',
              optimizer = 'Adam',
              metrics = ['accuracy'])
```

Gambar 9. Uji Akurasi

```
history = model.fit(
    train_generator,
    #steps_per_epoch=8,
    epochs = 50,
    validation_data = val_generator,
    validation_steps = 1,
    verbose = 1,
    callbacks = [callbacks]
)
```

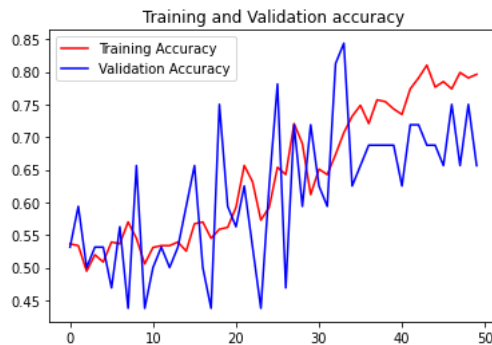
Gambar 10. Epoch

```
Epoch 50/50
12/11 [-----] - 175 15/step - loss: 0.4559 - accuracy: 0.7961 - val_loss: 0.4718 - val_accuracy: 0.6562
```

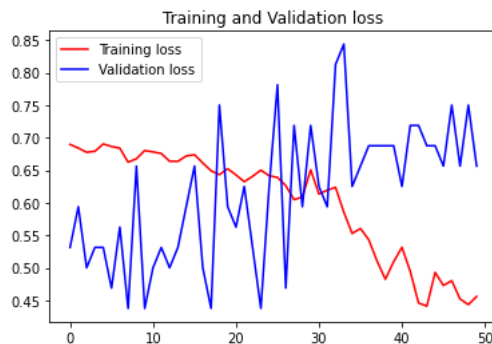
Gambar 11. Hasil Akurasi

Berdasarkan gambar 12 yang menunjukkan hasil akurasi model, dapat diketahui besar loss 0,45% dengan akurasi yang cukup baik 79,61%. Sedangkan untuk data validasi, nilai loss rendah yaitu sebesar

47% dan akurasi sebesar 65,52%. untuk klasifikasi tersebut dapat dikatakan sudah bagus untuk dijalankan, dan diterapkan pada sistem deteksi helm *safety* secara *real-time*. Berikut ini akan dipaparkan gambar grafik akurasi dan loss dari model *training* dan validasi



Gambar 12. Grafik Model Akurasi



Gambar 13. Grafik Model Loss

D. Pengujian

Pada tahap ini dilakukan uji akurasi Pada tahap dalam mendeteksi objek. Pada penelitian ini, pengujian dilakukan menggunakan data uji yang telah disiapkan sebelumnya. Output pengujian ini diharapkan berupa kotak pembatas (bounding box) yang mengelilingi objek yang terdeteksi, baik yang menggunakan helm keselamatan maupun yang tidak terdeteksi. Hasil pengujian menunjukkan implementasi sistem pada area praktik dalam ruangan. Dari gambar yang disajikan, dapat disimpulkan bahwa sistem mampu melakukan deteksi secara *real-time* dengan baik, meskipun nilai keakuratan berubah sesuai dengan gerakan, pencahayaan, dan ketajaman gambar saat ditangkap oleh sistem. Dari hasil uji coba, tingkat keyakinan tertinggi yang dicapai adalah 97%, sedangkan nilai terendahnya adalah 62%.

```

python3 train.py --img_dir /home/.../img_dir --gt_dir /home/.../gt_dir --num_epochs 100 --batch_size 16 --num_workers 4 --device cuda:0 --save_dir /home/.../save_dir --resume /home/.../resume_dir

python3 test.py --img_dir /home/.../img_dir --gt_dir /home/.../gt_dir --num_epochs 100 --batch_size 16 --num_workers 4 --device cuda:0 --save_dir /home/.../save_dir --resume /home/.../resume_dir

python3 eval.py --img_dir /home/.../img_dir --gt_dir /home/.../gt_dir --num_epochs 100 --batch_size 16 --num_workers 4 --device cuda:0 --save_dir /home/.../save_dir --resume /home/.../resume_dir

python3 visualize.py --img_dir /home/.../img_dir --gt_dir /home/.../gt_dir --num_epochs 100 --batch_size 16 --num_workers 4 --device cuda:0 --save_dir /home/.../save_dir --resume /home/.../resume_dir

```

Gambar 14. Hasil Training Yolo



Gambar 15. Hasil Deteksi Yolo

E. Evaluasi Model

Pada tahap ini dilakukan uji akurasi Performa model Convolutional Neural Network untuk klasifikasi penggunaan helm keselamatan harus dievaluasi untuk mengetahui seberapa baik prediksinya terhadap data uji. Penelitian ini menggunakan metode 10-fold cross-validation. Skenario pertama dilakukan untuk mencari epoch optimal bagi model dalam tahap pelatihan agar memperoleh performa terbaik. Epoch adalah satu putaran di mana seluruh dataset telah melewati tahap pelatihan dalam Neural Network. Penelitian ini menguji beberapa nilai epoch, yaitu 10, 20, 30, 40, dan 50. Pada pengujian epoch, ukuran batch yang digunakan adalah 32 dan menggunakan Adam optimizer. Berikut adalah hasil pengujian nilai precision, accuracy, recall, dan F1 score pada setiap epoch.

Table I. HASIL EPOCH

Epoch	Akurasi	presisi	recal	F1 Score
10	0.7	0.75	0.7	0.71
20	0.7	0.78	0.77	0.78
30	0.7	0.8	0.77	0.77
40	0.79	0.8	0.79	0.8
50	0.8	0.83	0.8	0.8

Secara keseluruhan, hasil penelitian ini menunjukkan bahwa pengujian model YOLOv3-Tiny untuk deteksi pemakaian helm keselamatan mengindikasikan bahwa model ini layak diimplementasikan. Hal ini didasarkan pada hasil metrik kinerja yang menunjukkan akurasi, presisi, recall, dan F1 score yang konsisten tinggi, semuanya di atas 70%, bahkan mencapai 80% pada epoch ke-50. Dalam aplikasi machine learning, model dengan nilai presisi dan recall sekitar 80% umumnya dianggap cukup baik

Dari segi performa, model ini menunjukkan peningkatan akurasi seiring dengan bertambahnya jumlah epoch selama proses pelatihan. Pada epoch ke-50, YOLOv3-Tiny mencapai performa terbaiknya dengan akurasi 0.80, presisi 0.83, recall 0.80, dan F1 score 0.80, yang menunjukkan bahwa model berhasil menemukan keseimbangan optimal dalam mengenali objek helm keselamatan secara konsisten. Hasil ini menegaskan bahwa epoch ke-50 merupakan jumlah iterasi yang ideal untuk memaksimalkan kinerja deteksi objek dalam skenario real-time. Selain itu, dalam uji real-time, nilai keyakinan deteksi tertinggi mencapai 97%, sementara nilai terendah berada di angka 62%. Rentang keyakinan ini menunjukkan bahwa model

mampu mengenali objek dengan baik dan cukup stabil, meskipun keyakinan sedikit menurun pada kondisi pencahayaan yang bervariasi dan pada sudut tertentu.

IV. KESIMPULAN

Pada pengujian YOLOv3-Tiny untuk menilai akurasi model dalam mendeteksi objek, khususnya penggunaan helm safety. Pengujian ini dilakukan dengan data uji yang telah disiapkan, dimana output yang diharapkan adalah kotak pembatas yang mengelilingi objek yang terdeteksi. Dari hasil pengujian, sistem mampu melakukan deteksi secara real-time dengan baik dalam area praktik di dalam ruangan. Deteksi tetap stabil meskipun terdapat variasi dalam gerakan, pencahayaan, dan ketajaman gambar. Berdasarkan hasil uji coba, nilai keyakinan tertinggi dari model mencapai 97%, sementara nilai keyakinan terendah berada di angka 62%, menunjukkan kemampuan model untuk mengenali objek dengan tingkat keyakinan yang cukup baik. Pada model performa YOLOv3-Tiny dalam mendeteksi pemakaian helm safety pada skenario real-time menunjukkan hasil yang memadai. Model ini menunjukkan peningkatan performa seiring bertambahnya jumlah epoch, sebagaimana terlihat pada peningkatan nilai akurasi, presisi, recall, dan F1 score. Pada epoch ke-50, model mencapai performa terbaiknya, dengan akurasi sebesar 0.80, presisi 0.83, recall 0.80, dan F1 score juga 0.80. Hal ini menandakan bahwa pada epoch ini, model YOLOv3-Tiny mampu mengenali dan mendeteksi objek helm dengan akurasi yang tinggi secara real-time. Dengan demikian, epoch ke-50 dapat dianggap sebagai parameter optimal untuk mencapai performa deteksi terbaik pada skenario ini.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada seluruh pihak yang telah mendukung penulis dalam penulisan ini.

DAFTAR PUSTAKA

- [1] W. M. Wisnu, M. Imam, and Z. M. Syarifuz, "Pengolahan Citra Digital Pada Perhitungan Ikan Hias Menggunakan Metode Blob," *JRKTL. Jurnal Riset Kajian Teknologi & Lingkungan*, vol. 6, no. 2, pp. 118–126, 2023.
- [2] A. R. Putri, "Pengolahan Citra Dengan Menggunakan Web Cam Pada Kendaraan Bergerak Di Jalan Raya," *JUPI. Jurnal Ilmiah Pendidikan Informatika*, vol. 1, no. 1, pp. 1–6, 2016.
- [3] M. Effendi, Fitriyah, and U. Effendi, "Identifikasi Jenis dan Mutu Teh Menggunakan Pengolahan Citra Digital dengan Metode Jaringan Syaraf Tiruan," *J. Teknotan*, vol. 11, no. 2, pp. 67–76, Oct. 2017.
- [4] A. Jonathan and I. Wasito, "Perancangan Aplikasi Pengenalan Aksara Jawa Digital Menggunakan Convolutional Neural Network dan Computer Vision," *J. Pendidikan Teknologi Informasi*, vol. 3, no. 2, pp. 364–377, Aug. 2023.
- [5] T. W. Qur'ana, "Implementasi Metode Convolutional Neural Network (CNN) untuk Klasifikasi Motif pada Citra Sasirangan," *J. Ilmiah Multidisiplin*, vol. 1, no. 7, pp. 647–653, 2023.
- [6] N. Hardi and J. Sundari, "Pengenalan Telapak Tangan Menggunakan Convolutional Neural Network (CNN)," *J. Rekayasa Perangkat Lunak*, vol. 4, no. 1, pp. 10–15, 2022.
- [7] Y. Hartiwi, E. Rasywir, Y. Pratama, and P. A. Jusia, "Eksperimen Pengenalan Wajah dengan fitur Indoor Positioning System menggunakan Algoritma CNN," *J. Informatika dan Komputer*, vol. 22, no. 2, pp. 109–116, 2020.
- [8] B. Yanto, L. Fimawahib, A. Supriyanto, B. H. Hayadi, and R. R. Pratama, "Klasifikasi Tekstur Kematangan Buah Jeruk Manis Berdasarkan Tingkat Kecerahan Warna dengan Metode Deep Learning Convolutional Neural Network," *J. Inovtek Polbeng - Seri Informatika*, vol. 6, no. 2, pp. 259–268, 2021.
- [9] E. A. Nour and Harintaka, "Implementasi Metode Convolutional Neural Network Untuk Klasifikasi Tanaman Pada Citra Resolusi Tinggi," *J. Geomatika*, vol. 24, no. 2, pp. 61–68, Nov. 2018.

-
- [10] X. Zhang, W. Yang, X. Tang, and J. Liu, "A fast learning method for accurate and robust lane detection using two-stage feature extraction with YOLO v3," *J. Sensors (Switzerland)*, vol. 18, no. 12, pp. 1–20, Dec. 2018.
- [11] M. Alfin Taufiqurrochman, H. Februriyanti, and J. Homepage, "Rancang Bangun Aplikasi Deteksi Alat Pelindung Diri (APD) untuk Pekerja Proyek dengan Menggunakan Algoritma YOLOv5 J. JTIK (Jurnal Teknologi Informasi dan Komunikasi)," *Jurnal Teknologi Informasi dan Komunikasi*, vol. 8, no. 2, p. 2024, 2024.
- [12] I. K. Nti, O. N. Boateng, and J. Aning, "Performance of Machine Learning Algorithms with Different K Values in K-fold CrossValidation," *J. International Journal of Information Technology and Computer Science*, vol. 13, no. 6, pp. 61–71, Dec. 2021.
- [13] L. Agustien, T. Rahman, and A. W. Hujairi, "Real-time Deteksi Masker Berbasis Deep Learning menggunakan Algoritma CNN YOLOv3," *J. Teknologi Informasi dan Terapan*, vol. 8, no. 2, pp. 129–137, 2021.
- [14] H. LiGochoo and Munkhjargal, "Safety Helmet Wearing Dataset," *J. Mendeley*, Apr. 2021.
- [15] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks," Dec. 2013.
- [16] M. F. Arif, A. Nurkholis, S. Laia, and P. Rosyani, "Deteksi Kendaraan Dengan Metode YOLO," *J. Artificial Intelligent dan Sistem Penunjang Keputusan*, vol. 01, no. 01, pp. 1–8, 2023.
- [17] M. O. Lawal, "Tomato detection based on modified YOLOv3 framework," *J. Scientific Reports*, vol. 11, no. 1, Dec. 2021.