

Deteksi Objek Pisau Menggunakan Yolo *Machine Learning*

Andreas Restu Priatama
Teknik Informatika, Program Pascasarjana, Universitas Pamulang
e-mail: andreas.restu.p@gmail.com

Abstrak—Perkembangan teknologi telah menjadi pendorong utama dalam kemajuan pesat dalam bidang *machine learning*. Dengan pertumbuhan komputasi yang kuat dan akses ke data yang melimpah, *machine learning* telah mengalami perkembangan yang signifikan dalam beberapa dekade terakhir. Teknologi komputasi yang semakin kuat telah memungkinkan pelatihan model *deep learning* yang lebih kompleks. Penggunaan unit pemrosesan grafis (GPU) dan penggunaan teknologi awan telah mempercepat pelatihan model dan memungkinkan penggunaan arsitektur jaringan saraf yang lebih mendalam. Teknologi sensor yang canggih, termasuk kamera, sensor citra, dan teknologi pengolahan citra, telah menghasilkan data berkualitas tinggi yang mendukung deteksi sebuah objek yang lebih akurat. Sensor-sensor ini mampu mengidentifikasi objek dalam berbagai kondisi pencahayaan dan sudut pandang. Penelitian ini menekankan bahwa perkembangan teknologi telah membawa perubahan positif dalam deteksi objek yang dalam hal ini adalah deteksi pisau, menciptakan solusi yang lebih efisien dan andal untuk meningkatkan keamanan masyarakat dan lingkungan. Penelitian ini mengaplikasikan deteksi objek menggunakan Yolo yang digunakan untuk mendeteksi pisau dalam berbagai konteks termasuk mempertimbangkan pendekatan konvensional dan solusi berbasis teknologi terkini, serta penggunaan *machine learning* dan *deep learning*. Penelitian ini juga membahas *platform* Yolo dalam mendeteksi sebuah objek.

Kata Kunci— Deteksi Pisau; *Machine Learning*; Neural Network; Yolo; Kecerdasan Buatan.

I. PENDAHULUAN

Perkembangan teknologi telah membawa perubahan mendasar dalam dunia kecerdasan buatan (*Artificial Intelligence*, AI). Seiring berjalannya waktu, AI telah menjadi salah satu bidang yang paling dinamis dan inovatif di dunia teknologi. Dari konsep awalnya pada tahun 1950-an hingga saat ini, AI telah mengalami transformasi radikal, didorong oleh kemajuan dalam berbagai teknologi yang mendukungnya. Pendahuluan ini akan membahas bagaimana perkembangan teknologi telah memengaruhi dan membentuk perkembangan AI, serta bagaimana interaksi yang erat antara keduanya telah membuka pintu menuju berbagai aplikasi yang mendalam di berbagai sektor [6].

Perkembangan teknologi komputasi, terutama pemrosesan grafis dan komputasi paralel, telah memungkinkan pelatihan model-model AI yang lebih kompleks dan akurat. GPU (*Graphics Processing Unit*) dan arsitektur pemrosesan yang lebih efisien telah mempercepat proses pembelajaran mesin (*machine learning*) dan *deep learning*, yang menjadi dasar kebanyakan aplikasi AI saat ini. Ini telah memungkinkan penggunaan AI dalam masalah yang lebih rumit, seperti pengenalan wajah, pengolahan bahasa alami, dan visi komputer yang lebih canggih. Selain itu, perkembangan dalam teknologi penyimpanan data dan kapasitas komputasi awan telah memungkinkan akses mudah ke sumber daya komputasi yang kuat secara fleksibel. Ini telah mendukung pengembangan aplikasi kecerdasan buatan yang terdistribusi dan beroperasi dalam waktu nyata, seperti layanan berbasis *cloud* yang menyediakan kemampuan kecerdasan buatan yang lebih luas kepada pengguna akhir.

Deteksi objek dalam gambar merupakan salah satu aplikasi paling menonjol dalam dunia *machine learning* dan visi komputer. Ini memungkinkan komputer untuk mengenali dan lokalisasi objek-objek dalam gambar, yang memiliki banyak aplikasi praktis, mulai dari pengawasan keamanan hingga mobil otonom. Dalam penelitian ini, akan menjelaskan konsep dasar deteksi objek pisau menggunakan *machine learning Convolutional Neural Network*. [1]

Penelitian mengenai deteksi pisau sebelumnya telah dilakukan oleh Ali Mahmudi yang berjudul “DETEKSI SENJATA TAJAM DENGAN METODE HAAR CASCADE CLASSIFIER MENGGUNAKAN TEKNOLOGI SMS GATEWAY”. Perbedaan penelitian tersebut dengan penelitian ini adalah pada metode *machine learning* yang digunakan. Pada penelitian sebelumnya digunakan metode *Haar Cascade*, sedangkan dalam penelitian ini menggunakan *platform* YOLO yang menggunakan metode *Convolutional Neural Network* (CNN).

Penelitian ini akan menjelaskan secara rinci tentang bagaimana teknologi, khususnya pendekatan YOLO, digunakan dalam deteksi objek pisau, yang akan menjelaskan konsep dasar, tahap pengembangan, dan tantangan yang terkait dengan penerapan teknologi ini dalam konteks keamanan dan pengawasan, serta potensi aplikasi dan dampak positif yang mungkin dihasilkan dari kemajuan ini.

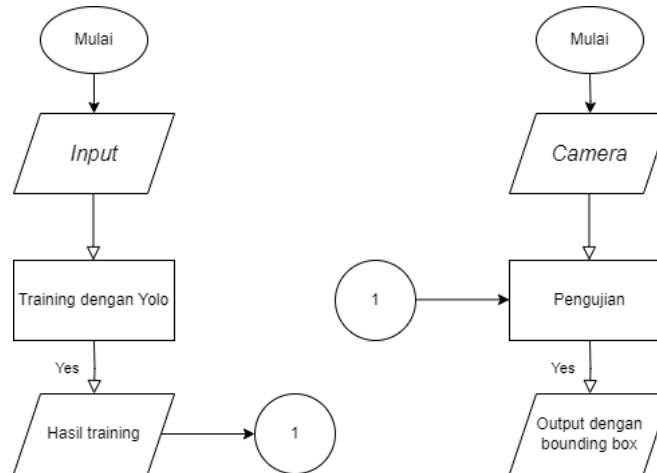
II. METODE PENELITIAN

A. YOLO

Salah satu *platform* kecerdasan buatan yang sering digunakan dalam bidang deteksi objek menggunakan *machine learning* adalah YOLO (You Only Look Once). YOLO adalah sebuah pendekatan yang dirancang untuk mendeteksi objek dalam citra

atau video dalam waktu nyata. Pendekatan ini pertama kali diusulkan oleh Joseph Redmon, Santosh Divvala, Ross Girshick, dan Ali Farhadi pada tahun 2016, dan sejak itu telah menjadi salah satu metode deteksi objek yang paling populer dan efisien. Konsep dasar di balik YOLO adalah menggabungkan tahap deteksi objek dan klasifikasi objek menjadi satu proses yang berjalan bersamaan. Hal ini berbeda dari pendekatan konvensional yang memerlukan beberapa tahap pemrosesan berurutan, seperti pemilihan wilayah berbasis pemotongan citra (*region-based image cropping*) dan pemrosesan ulang. Dengan YOLO, model dapat secara langsung mengeluarkan lokasi objek dan label kelasnya dalam satu perangkat keras. YOLO telah digunakan dalam berbagai aplikasi, seperti pemantauan keamanan, kendaraan otonom, pengenalan wajah, dan banyak lagi.[1]

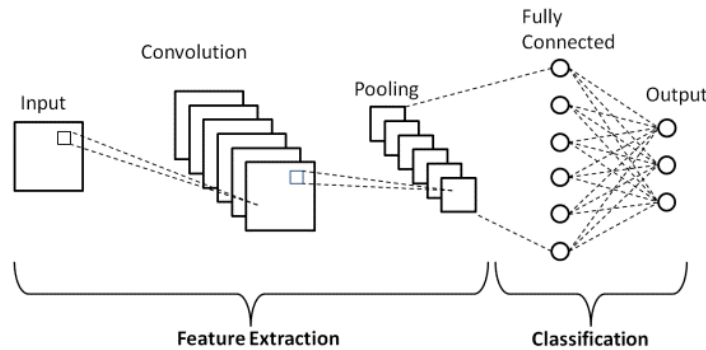
Salah satu penerapan YOLO dalam bidang keamanan adalah deteksi senjata tajam (pisau).Keunggulan dari deteksi senjata tajam dengan *machine learning* adalah kemampuannya untuk memberikan respons cepat dan otomatis ketika senjata tajam terdeteksi. Ini memiliki aplikasi potensial dalam berbagai konteks, seperti keamanan di bandara, pengawasan kawasan publik, atau keamanan perangkat lunak. Selain itu, perkembangan dalam teknologi sensor, seperti kamera beresolusi tinggi, telah mendukung pengumpulan data citra berkualitas tinggi yang mendukung pelatihan model *machine learning*. Ini memungkinkan model-model *machine learning* untuk memahami karakteristik senjata tajam dengan lebih baik dan meningkatkan akurasi deteksinya.



Gambar 1.
Flowchart Proses Yolo

B. Convolutional Neural Network

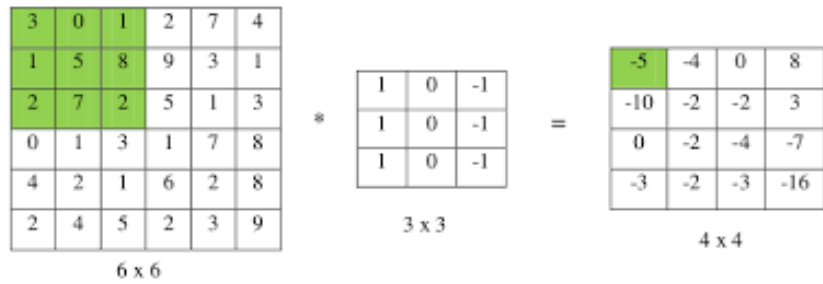
Convolutional Neural Network (CNN) adalah jenis arsitektur jaringan saraf tiruan yang dikhususkan untuk tugas-tugas pengolahan citra dan pengenalan pola visual. CNN dirancang untuk mengekstraksi fitur-fitur penting dari data citra atau grid data lainnya, seperti pengenalan objek dalam gambar. CNN memiliki beberapa komponen penting yang membentuk strukturnya yaitu *convolution layer*, *pooling layer*, dan *fully connected layer*.[1]



Gambar 2.
Struktur Convolutional Neural Network (upgrad.com)

1) Convolutional Layer

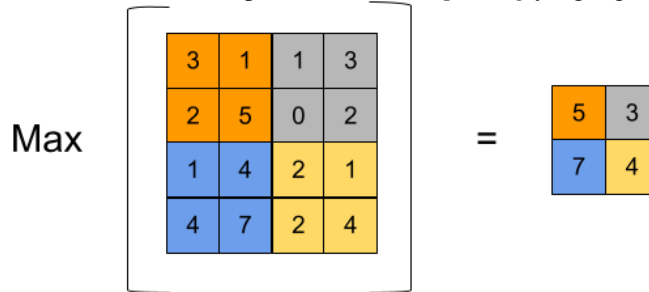
Convolutional Layer adalah salah satu elemen dasar *Convolutional Neural Network*, Convolution Layer melakukan operasi konvolusi pada output dari layer sebelumnya. Layer tersebut adalah proses utama yang mendasari sebuah CNN. Konvolusi adalah suatu istilah matematis yang berarti mengaplikasikan sebuah fungsi pada *output* fungsi lain secara berulang. Dalam pengolahan citra, konvolusi berarti mengaplikasikan sebuah kernel pada citra di semua *offset* yang memungkinkan. Kotak hijau secara keseluruhan adalah citra yang akan dikonvolusi. Kernel bergerak dari sudut kiri atas ke kanan bawah. Sehingga hasil konvolusi dari citra tersebut dapat dilihat pada gambar disebelah kanannya. Tujuan dilakukannya konvolusi pada data citra adalah untuk mengekstraksi fitur dari citra input. Konvolusi akan menghasilkan transformasi linear dari data input sesuai informasi spasial pada data. Bobot pada layer tersebut menspesifikasikan kernel konvolusi yang digunakan, sehingga kernelkonvolusi dapat dilatih berdasarkan input pada CNN. [2]



Gambar 3.
 Convolutional Layer[3]

2) *Pooling Layer*

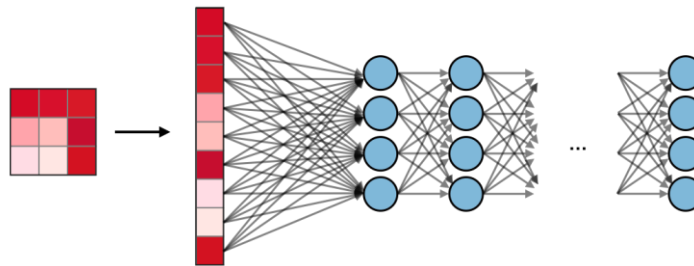
Pooling merupakan pengurangan ukuran matriks dengan menggunakan operasi *pooling*. *Pooling layer* biasanya berada setelah lapisan konvolusi. Pada dasarnya *pooling layer* terdiri dari sebuah filter dengan ukuran dan *stride* tertentu yang akan secara bergantian bergeser pada seluruh area *feature map*. Dalam *pooling layer* terdapat dua macam *pooling* yang biasa digunakan yaitu *average pooling* dan *max-pooling*. Nilai yang diambil pada *average pooling* adalah nilai rata-rata, sedangkan pada *max-pooling* adalah nilai maksimal. Lapisan *Pooling* yang dimasukkan diantara lapisan konvolusi secara berturut-turut dalam arsitektur model CNN dapat secara progresif mengurangi ukuran *volume output* pada *feature map*, sehingga mengurangi jumlah parameter dan perhitungan di jaringan, untuk mengendalikan *overfitting*. Lapisan *pooling* bekerja di setiap tumpukan *feature map* dan melakukan pengurangan pada ukurannya. Bentuk lapisan *pooling* umumnya dengan menggunakan filter dengan ukuran 2x2 yang diaplikasikan dengan langkah sebanyak dua dan beroperasi pada setiap irisan dari inputnya [3]. Di bawah ini merupakan contoh *max pooling* yang digunakan dalam CNN.



Gambar 4.
 Pooling Layer (socs.binus.ac.id)

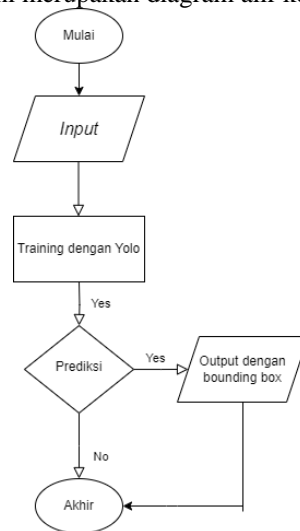
3) *Fully Connected Layer*

Fully Connected Layer adalah lapisan di mana semua neuron aktivasi dari lapisan sebelumnya terhubung semua dengan neuron di lapisan selanjutnya seperti halnya jaringan syaraf tiruan biasa. Setiap aktivasi dari lapisan sebelumnya perlu diubah menjadi data satu dimensi sebelum dapat dihubungkan ke semua neuron di *Fully Connected Layer*. *Fully Connected Layer* dilakukan setelah melewati proses *Convolutional Layer* dan *Pooling Layer*. Perbedaan antara *Fully Connected Layer* dan lapisan *Convolutional Layer* adalah neuron di lapisan konvolusi terhubung hanya ke daerah tertentu pada input, sedangkan *Fully Connected Layer* memiliki neuron yang secara keseluruhan terhubung. Namun, kedua lapisan tersebut masih mengoperasikan produk dot, sehingga fungsinya tidak begitu berbeda. *Fully Connected Layer* pada intinya adalah sebuah neural network multilayer perceptron (MLP), yang memiliki beberapa hidden layer, activation function, output layer dan loss function. *Fully connected layer* lah yang akan berperan untuk mengklasifikasi data masukan [5].



Gambar 5.
 Fully connected layer

Berdasarkan beberapa tahapan diatas, berikut ini merupakan diagram alir keseluruhan dari sistem yang akan dibuat.



Gambar 6.
Flowchart Keseluruhan Sistem

III. HASIL DAN PEMBAHASAN

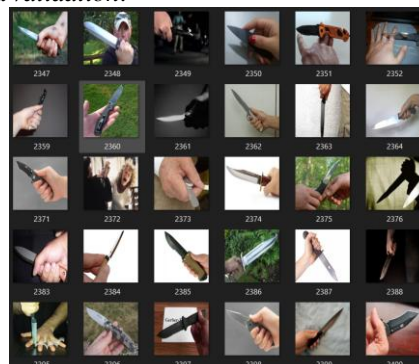
Proses dimulai dengan mengumpulkan data, kemudian melakukan pra-pemrosesan data, *training* data dan pengujian data. Adapaun penjelasan detail mengenai hasil dari deteksi, dan gambar proses secara keseluruhan dapat dilihat di bawah ini pada Gambar 5.

A. Pengumpulan data

Dalam konteks YOLO (*You Only Look Once*) atau pemrosesan *deep learning* lainnya, data training dan data validasi adalah dua set data yang digunakan untuk melatih dan menguji model deteksi objek. Ini adalah bagian penting dalam pengembangan model deteksi objek yang akurat. Berikut adalah penjelasan singkat tentang kedua konsep ini.

Data *training* adalah set data yang digunakan untuk melatih model deteksi objek, termasuk model YOLO. Set ini biasanya berisi citra atau video yang telah di-annotasi atau diberi label dengan informasi mengenai lokasi objek yang ingin dideteksi (dalam hal ini, pisau) beserta label kelas yang sesuai. Proses pelatihan model melibatkan penyajian data training kepada model, sehingga model dapat mempelajari pola dan fitur yang diperlukan untuk mendeteksi objek dengan akurasi. Data training harus mencakup beragam situasi, seperti berbagai sudut pandang, pencahayaan, dan kondisi latar belakang, untuk memastikan model mampu mengenali objek dengan baik dalam berbagai situasi. Sedangkan data validasi adalah set data yang digunakan untuk menguji kinerja model yang telah dilatih dengan data training. Set data ini seringkali terpisah dan berbeda dari data training untuk menghindari *overfitting* (kecenderungan model untuk mengingat data training dengan sangat baik, tetapi tidak dapat melakukan generalisasi dengan baik pada data baru). Dalam konteks YOLO, data validasi digunakan untuk mengukur akurasi model, menghitung metrik seperti tingkat deteksi yang benar, tingkat kesalahan, dan akurasi klasifikasi. Data validasi membantu menilai apakah model bekerja dengan baik dalam situasi yang berbeda.

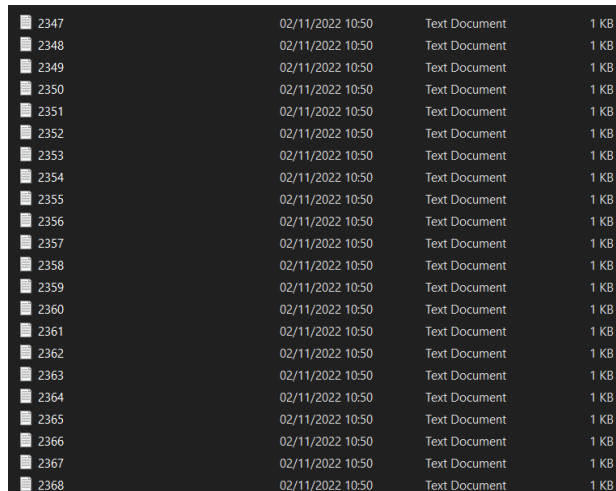
Untuk melakukan sebuah training di perlukan data gambar yang tepat, untuk mendeteksi pisau secara akurat ada beberapa hal yang harus diperhatikan salah satunya situasi pencahayaan rendah atau dalam berbagai sudut pandang. Identifikasi sumber data yang relevan dilakukan dengan cara mengumpulkan citra yang mengandung gambar pisau dari berbagai sumber, seperti internet, perangkat kamera, atau sumber data yang relevan lainnya. Total data citra yang digunakan dalam deteksi pisau pada penelitian ini berjumlah 600 data *training* dan 100 data *validation*.



Gambar 7.
Data Citra Pisau

B. Pra-Pemrosesan

Normalisasi data adalah proses mengubah data gambar ke dalam format yang seragam, sehingga nilai piksel dalam citra berada dalam rentang tertentu, seperti antara 0 dan 1. Hal ini membantu model belajar dengan lebih baik. Normalisasi juga bisa berarti mengubah ukuran citra menjadi ukuran yang konsisten, misalnya 416x416 piksel, yang sering digunakan dalam model YOLO. Setelah mengumpulkan dataset, selanjutnya adalah melakukan anotasi data yaitu menandai atau memberikan label pada citra atau video untuk menunjukkan lokasi dan jenis objek yang ingin dideteksi (objek pisau) menggunakan labelImg. Anotasi mencakup informasi tentang koordinat relatif objek di dalam citra, seperti titik tengah, tinggi, dan lebar objek. Dataset Anda perlu dibagi menjadi dua bagian: data pelatihan (*training data*) dan data validasi (*validation data*) baik dari gambar ataupun labelnya.

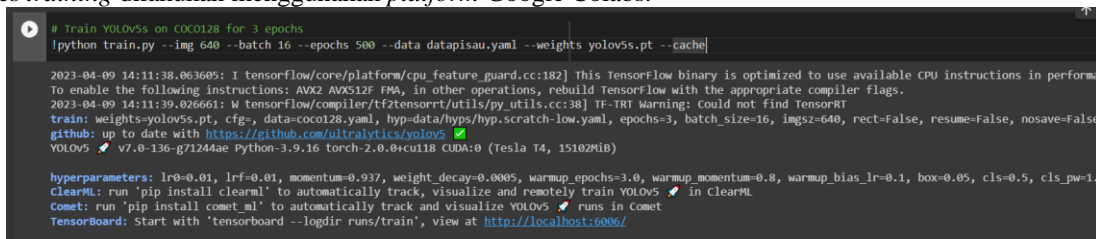


2347	02/11/2022 10:50	Text Document	1 KB
2348	02/11/2022 10:50	Text Document	1 KB
2349	02/11/2022 10:50	Text Document	1 KB
2350	02/11/2022 10:50	Text Document	1 KB
2351	02/11/2022 10:50	Text Document	1 KB
2352	02/11/2022 10:50	Text Document	1 KB
2353	02/11/2022 10:50	Text Document	1 KB
2354	02/11/2022 10:50	Text Document	1 KB
2355	02/11/2022 10:50	Text Document	1 KB
2356	02/11/2022 10:50	Text Document	1 KB
2357	02/11/2022 10:50	Text Document	1 KB
2358	02/11/2022 10:50	Text Document	1 KB
2359	02/11/2022 10:50	Text Document	1 KB
2360	02/11/2022 10:50	Text Document	1 KB
2361	02/11/2022 10:50	Text Document	1 KB
2362	02/11/2022 10:50	Text Document	1 KB
2363	02/11/2022 10:50	Text Document	1 KB
2364	02/11/2022 10:50	Text Document	1 KB
2365	02/11/2022 10:50	Text Document	1 KB
2366	02/11/2022 10:50	Text Document	1 KB
2367	02/11/2022 10:50	Text Document	1 KB
2368	02/11/2022 10:50	Text Document	1 KB

Gambar 8.
Data Pra-Pemrosesan

C. Training

Pelatihan data dalam YOLO adalah salah satu langkah kunci dalam pengembangan model deteksi objek yang akurat. Dalam konteks YOLO, data pelatihan mengacu pada set data yang digunakan untuk melatih model YOLO agar dapat mengenali dan mendeteksi objek dengan akurasi. perlu menyesuaikan *hyperparameter* seperti *batch size*, dan *epoch* dalam model YOLO untuk meningkatkan kinerja model. Dalam percobaan pertama *training setting hyperparameter* yang diberikan yaitu 16 *batch* dan 500 *epochs*. Proses *training* dilakukan menggunakan *platform* Google Colabs.



```
# Train YOLOv5s on COCO128 for 3 epochs
python train.py --img 640 --batch 16 --epochs 500 --data datapisau.yaml --weights yolov5s.pt --cache

2023-04-09 14:11:38.063605: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use available CPU instructions in performance. To enable the following instructions: AVX2 AVX512F FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
2023-04-09 14:11:39.026661: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not find TensorRT
train: weights=yolov5s.pt, cfg=, data=coco128.yaml, hyp=data/hyps/hyp.scratch-low.yaml, epochs=3, batch_size=16, imgsz=640, rect=False, resume=False, nosave=False
github: up to date with https://github.com/ultralytics/yolov5
YOLOv5 v7.0-136-g71244ae Python-3.9.16 torch-2.0.0+cu118 CUDA:0 (Tesla T4, 15102MiB)

hyperparameters: lr0=0.01, lrf=0.01, momentum=0.937, weight_decay=0.0005, warmup_epochs=3.0, warmup_momentum=0.8, warmup_bias_lr=0.1, box=0.05, cls=0.5, cls_pw=1.0
ClearML: run 'pip install clearml' to automatically track, visualize and remotely train YOLOv5 in ClearML
Comet: run 'pip install comet_ml' to automatically track and visualize YOLOv5 runs in Comet
TensorBoard: Start with 'tensorboard --logdir runs/train', view at http://localhost:6006/
```

Gambar 9.
Proses Training Google Colab

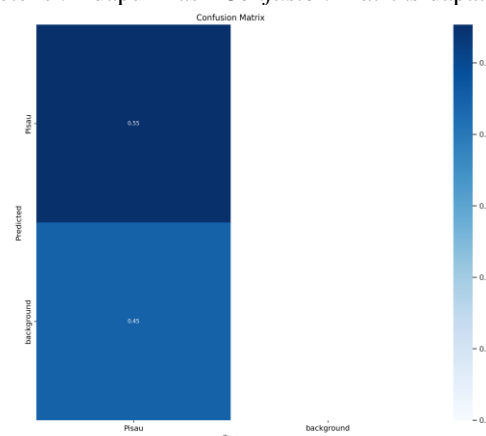
D. Pengujian

Pengujian diawali dengan menginstalasi semua pustaka dan alat pendukung yang diperlukan, seperti OpenCV, numpy, dan pustaka YOLO yang sesuai. Selain itu dibutuhkan webcam atau kamera eksternal yang terhubung ke komputer. Setelah dilakukan training dari data menghasilkan mAP (Mean Average Precision) sebesar 56% mungkin kurang memuaskan jika Anda berusaha untuk aplikasi yang sangat kritis, seperti kendaraan otonom yang memerlukan tingkat akurasi yang lebih tinggi. Ada beberapa cara untuk meningkatkan mAP, termasuk melatih lebih lama, menggunakan dataset yang lebih besar, mengubah arsitektur model, atau melakukan data augmentation. Berdasarkan hasil tersebut artinya 600 data training masih sangatlah kurang, sehingga diperlukan lebih banyak variasi data, untuk memperbanyak variasi data, salah satu metode yang dapat digunakan adalah augmentation. Penggunaan augmentasi data memungkinkan model machine learning untuk melihat lebih banyak variasi dalam data pelatihan, yang pada gilirannya dapat meningkatkan kemampuan model untuk mengenali objek atau pola dalam situasi nyata yang berbeda. Augmentasi adalah alat yang sangat berguna dalam mengatasi masalah overfitting, khususnya ketika dataset pelatihan terbatas atau tidak mencakup semua kasus yang mungkin dihadapi oleh model saat digunakan secara praktis. Adapun hasil dari pengujian dapat dilihat pada Gambar 8, dan Confusion Matrixnya dapat dilihat pada Gambar 9.



Gambar 10.
Hasil Deteksi

Dari hasil yang telah didapatkan, untuk melihat tingkat akurasi digunakan sebuah *confusion matrix* yang memuat informasi mengenai jumlah objek yang berhasil terdeteksi. Adapun hasil *Confusion Matrix* dapat dilihat pada Gambar 11 berikut.



Gambar 11.
Confusion Matrix

IV. KESIMPULAN

Berdasarkan hasil training yang telah dilakukan menggunakan Yolo untuk deteksi pisau, didapatkan kesimpulan bahwa YOLO adalah salah satu metode yang kuat dan efisien untuk deteksi objek dalam citra atau video. Ini telah terbukti memberikan akurasi deteksi objek yang tinggi, termasuk dalam kasus deteksi objek seperti pisau. Kesuksesan deteksi pisau dengan YOLO sangat tergantung pada data pelatihan yang baik. Pengumpulan dataset yang representatif dan pelatihan model dengan data tersebut adalah langkah kunci dalam mencapai hasil yang akurat.

YOLO memerlukan daya komputasi yang cukup besar, terutama ketika dijalankan pada video atau dalam lingkungan real-time. Penggunaan perangkat keras yang kuat atau implementasi yang dioptimalkan dapat membantu. Pengujian dengan jumlah data 600 menghasilkan akurasi sebesar 56%. Berdasarkan persentase akurasi yang dihasilkan masih terbilang rendah bisa jadi dikarenakan oleh jumlah dataset yang terbatas.

DAFTAR PUSTAKA

- [1] Qurotul Aini, Ninda Lutfiani, Hendra Kusumah, Muhammad Suzaki Zahran, "Deteksi dan Pengenalan Objek Dengan Model Machine Learning: Model Yolo". CESS (Journal of Computer Engineering System and Science) (2021) Vol. 6 No. 2
- [2] Sashmita Anggeli, Kemal Ade Sekarwati, "Implementasi Deep Learning Menggunakan Metode Convolutional Neural Network dan Multimedia Development Life Cycle pada Aplikasi Pengenalan Jenis Ikan Hias Berbasis Android", Pengembangan Rekayasa dan Teknologi, Vol 17, No.2, 2021.
- [3] Akhmad Rohim, Yuita Arum Sari, Tibyani, "Convolution Neural Network (CNN) Untuk Pengklasifikasian Citra Makanan Tradisional", Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, Vol. 3, No 7, 2019.
- [4] Andre Kanisius Edguard Lopian, Sherwin R.U.A Sompie, Pinrolinvic D. K. Manembu, "You Only Look Once (YOLO) Implementation For Signature Pattern Classification", Jurnal Teknik Informatika Vol. 16, 2021
- [5] Oektavia Nurima Putri, "Implementasi Metode Cnn Dalam Klasifikasi Gambar Jamur Pada Analisis Image Processing", TUGAS AKHIR, 2020
- [6] Alia Qonita Julia Selin, "Peranan Teknologi Artificial Intelligence di Era Revolusi Industri 4.0" Jurnal Sains dan Matematika, 2022.