1. ***Data Collection***

Data yang digunakan dalam penelitian ini merupakan kumpulan citra yang diperoleh dari *roboflow* dengan tipe *keypoint detection.* Data tersebut berjumlah 273 gambar dengan label “bad” sejumlah 176 dan “Good” sejumlah 97 data. Citra tersebut merepresentasikan posisi duduk yang paling sering dilakukan oleh orang dengan pengambilan gambar dengan sudut, posisi, dan juga pencahayaan yang berbeda. Sehingga, dataset cukup bervariasi untuk model deteksi posisi duduk.

1. ***Data Preprocessing***

Tahapan preprocessing data yang dilakukan dalam penelitian ini meliputi, *split dataset, auto-orient, resize,* dan *augmentasi.* Tahap preprocessing dilakukan dalam platform *roboflow* kecuali tahapan *augmentasi.* Tahapan *augmentasi* dilakukan langsung dengan python.

1. *Split Dataset*

Dalam *Deep learning,* proses *split dataset* merupakan proses membagi data menjadi beberapa bagian, train, validation, dan split. Proses yang dilakukan pada platform roboflow ini dimulai dengan mengunggah gambar yang sudah diberi label atau karena dataset sudah tersedia di roboflow maka bisa melakukan fork dataset pada data yang ada. Setelah dilakukan fork dataset, dilakukan split dataset secara acak dengan proporsi yang bisa disesuaikan. Pada penelitian ini proporsi split dataset yaitu, 70% (training), 20% (validasi), dan 10% (testing). Sehingga mendapatkan jumlah data train sebanyak 191 gambar, 55 gambar validasi, dan 27 gambar testing. Apabila pada saat training terdapat overitting/underfitting split dataset bisa disesuaikan lagi dengan menambah proporsi data validasi.

1. *Auto-orient*

Auto-orient merupakan fitur yang dimiliki oleh roboflow dengan tujuan untuk mengoreksi orientasi gambar untuk memastikan semua fambar memiliki orientasi yang konsisten secara otomatis. Meskipun penelitian ini fokus pada keypoint, namun auto-orient gambar juga diperlukan karena pada model yolo tetap memiliki bounding-box untuk memudahkan memberikan informasi yang dideteksi.

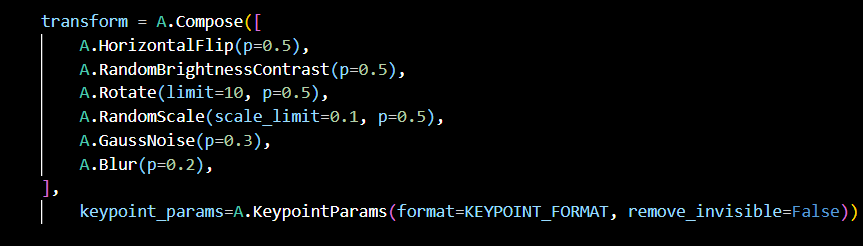
1. *Resize*

Model Yolov8s-pose yang digunakan dalam penelitian ini menggunakan ukuran gambar default yaitu 640x640 pixel. *Resize* yang digunakan dalam dataset ini yaitu fit within, yaitu mengubah dimensi gambar namun diisi dengan objek kosong seolah – olah gambar tetap dengan dimensi yang sama. Selain itu, apabila gambar diubah dimensinya dengan scretch maka detail objek bisa saja hilang karena beberapa gambar memiliki rasio 16:9, karena pada saat scretch rasio gambar menjadi 1:1. Oleh karena itu, semua gambar pada dataset ini harus disesuaikan dimensinya dengan tujuan untuk mencegah distorsi objek dalam gambar. Sehingga hasil deteksi tetap bisa akurat dan representif terhadap aslinya.

1. *Augmentasi*

Proses augmentasi data dalam penelitian ini dilakukan untuk meningkatkan jumlah dan keragaman citra pelatihan, mengingat jumlah dataset asli relatif terbatas, yaitu sekitar 200 citra. Augmentasi bertujuan untuk meningkatkan kemampuan generalisasi model YOLOv8-Pose dalam mengenali dua kelas postur duduk, yakni good dan bad, pada berbagai kondisi lingkungan dan variasi subjek. Dengan menambahkan variasi dalam data pelatihan, model diharapkan mampu mengenali pola postur dengan lebih akurat dan robust terhadap perubahan pencahayaan, sudut pandang kamera, serta perbedaan individu pengguna.

Teknik augmentasi yang digunakan berbasis pustaka Albumentations, yang mendukung transformasi spasial dan fotometrik sekaligus mempertahankan konsistensi posisi keypoint untuk setiap citra. Transformasi yang diterapkan meliputi pemantulan horizontal (horizontal flip), penyesuaian kecerahan dan kontras secara acak (random brightness and contrast adjustment), rotasi terbatas hingga ±10 derajat, skala acak (random scaling), penambahan derau (Gaussian noise), serta blur. Pemilihan jenis transformasi dilakukan dengan mempertimbangkan agar hasil augmentasi tetap mencerminkan kondisi realistis yang mungkin terjadi dalam konteks penggunaan sistem.

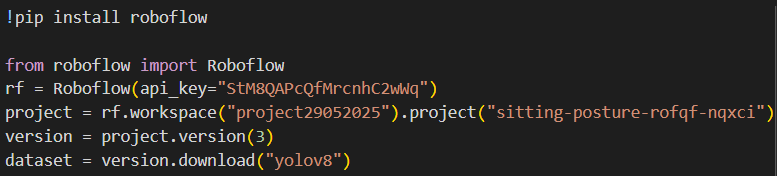


Setiap citra asli mengalami lima kali proses augmentasi dengan kombinasi transformasi yang dipilih secara acak, sehingga menghasilkan lima buah citra baru yang berbeda. Proses ini juga melibatkan penyesuaian ulang koordinat keypoint berdasarkan ukuran citra hasil augmentasi serta pelabelan ulang dalam format yang sesuai dengan standar YOLOv8-Pose, yaitu koordinat relatif (relative coordinates) dengan flag visibilitas. Hasil augmentasi kemudian disimpan dalam direktori tersendiri yang terpisah dari data asli, untuk kemudian digunakan sebagai data pelatihan utama. Dengan pendekatan ini, jumlah total data pelatihan meningkat menjadi sekitar 1200 citra, yang secara signifikan dapat memperkaya representasi data dan mendukung proses pelatihan model secara lebih optimal.

1. **Deteksi Posisi duduk Dengan yolov8s-pose**

Pada tahap ini, algoritma YOLOv8s-pose digunakan untuk melakukan deteksi keypoint. YOLOv8-pose merupakan versi yang lebih optimal dari metode YOLOv5-pose yang dikembangkan oleh *ultralytics* dan dikenal dengan kecepatan serta akurasi yang tinggi dalam mendeteksi objek maupun keypoint secara *realtime.* Sebelum masuk ke tahap training, yang perlu dilakukan adalah set up semua environment dan memastikan data sudah siap untuk digunakan. berikut adalah langkah – langkah set environtment yang dilakukan.

1. *Import Dataset*

Dataset yang digunakan pada penelitian ini diimport dari roboflow dan memastikan format yang digunakan sesuai dengan versi yang akan digunakan. Pada Gambar 1.2 format data yang digunakan yaitu yolov8 yang bisa dipilih di roboflow, kemudian roboflow akan memberikan code yang bisa digunakan untuk import dataset  


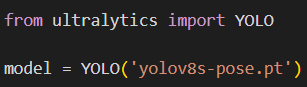
Pada Gambar 1.2 code sesuai dengan yang diberikan oleh roboflow dengan instalasi roboflow terlebih dahulu kemudian dilanjutkan dengan memasukan API key akun roboflow, workspae, dan project dengan versi yang diinginkan serta terdapat format dataset yang akan diimport.

1. Proses Augmentasi

Seperti yang sudah dijelaskan sebelumnya, proses augmentasi dilakukan langsung dengan python dengan trasnformasi yang sudah ditentukan

1. Setup model.

Sebelum memanggil model yang akan digunakan untuk training, pastikan library ultralitycs sudah terinstall. Jika sudah terinstall maka model dapat dipanggil untuk mendapatkan varian model yolov8s-pose seperti pada gambar 1.3 berikut.



Setelah mendapatkan model dengan format .pt dan model sudah berada pada folder, maka bisa lanjut ke tahap training.

1. Training model

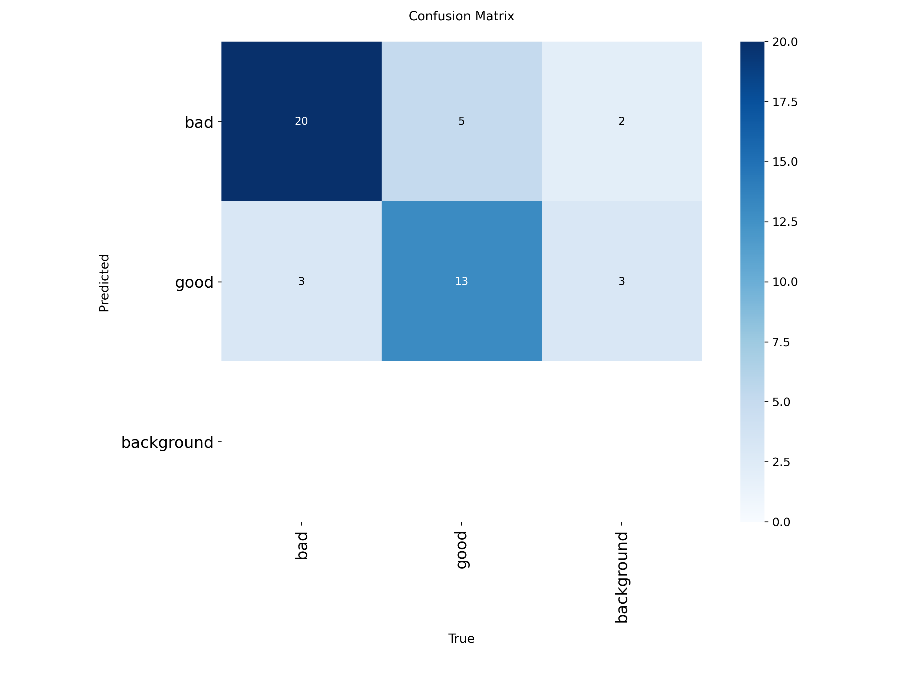
Sebelum melakukan training, pastikan data dan setup sudah sesuai dengan yang diinginkan. Selain itu, memastikan juga isi file data.yaml pada folder sitting-posture-3 sudah sesuai dengan folder yang akan digunakan untuk training, misalnya data training diubah yang awalnya berada di “sitting-posture-3” menjadi “augmented” karena data training sudah dilakukan augmentasi dan disimpan di folder yang berbeda. Jika semua sudah sesuai maka lanjut ke tahap training dengan code berikut.

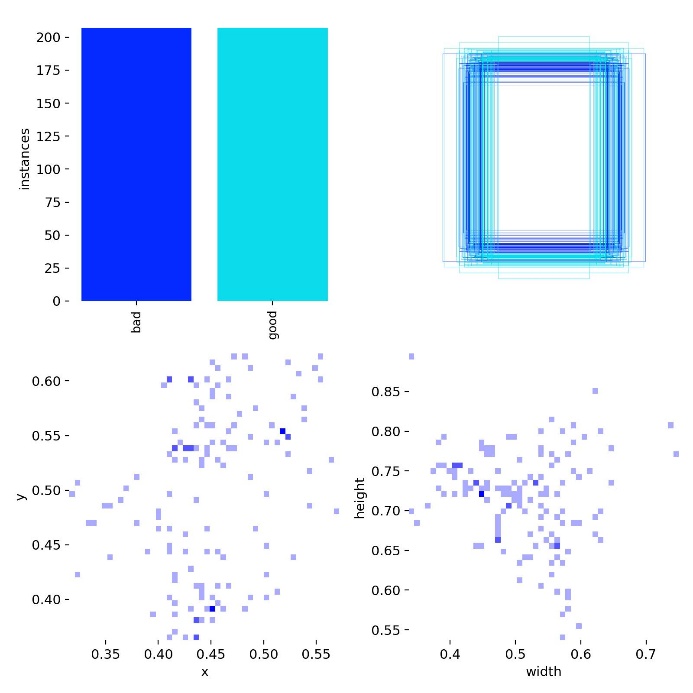
results = model.train(data="/content/sitting-posture-3/data.yaml", epochs=150, imgsz=640)

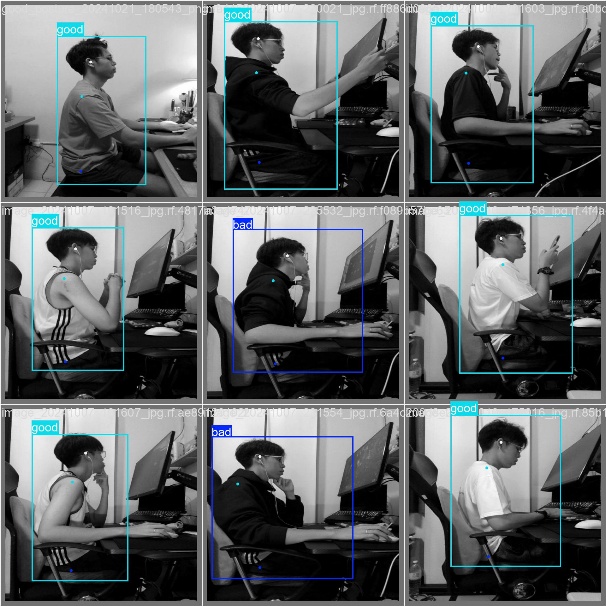
Code python tersebut sesuai dengan dokumentasi pada halaman web ultralitycs, yang dimana hanya mengubah epoch dan image size. Hyperparameter yang lain akan digunakan ketika hyperparameter standar yolo kesulitan mempelajari dataset posisi duduk. Hasil training dari YOLO akan tersimpan secara otomatis pada direktori berikut :



Pada direktori tersebut juga terdapat evaluasi model mulai dari grafik mAP dan loss, confusion matrix, hingga hasil deteksi dengan data test. Berikut adalah contoh beberapa hasil output yolov8s-pose selain model best.pt







1. **Evaluasi Model**

Evaluasi model dilakukan untuk mengukur performa deteksi postur duduk menggunakan arsitektur YOLOv8s-Pose yang telah dilatih pada dataset hasil augmentasi. Model yang digunakan memiliki 81 lapisan dan sejumlah 11.413.137 parameter, dengan kompleksitas komputasi sebesar 29.4 GFLOPs. Proses evaluasi dilakukan terhadap 55 citra pengujian yang mencakup 55 instance, terdiri atas dua kelas postur, yaitu Good dan Bad. Secara keseluruhan, model menunjukkan performa yang sangat baik dengan nilai presisi sebesar 0.929, recall sebesar 0.986, dan mean Average Precision (mAP) sebesar 0.987 untuk mAP@50 dan 0.743 untuk mAP@50–95 dalam evaluasi bounding box. Pada aspek deteksi pose (keypoint), model mencatatkan hasil yang identik untuk metrik presisi, recall, dan mAP@50, yaitu masing-masing sebesar 0.929, 0.986, dan 0.987, dengan mAP@50–95 sedikit lebih tinggi dibandingkan deteksi kotak, yaitu 0.987.

Jika dilihat berdasarkan masing-masing kelas, model menunjukkan performa yang sangat baik pada kelas Bad, dengan mAP@50 sebesar 0.99 dan mAP@50–95 sebesar 0.785 untuk deteksi kotak, serta nilai identik pada deteksi pose. Untuk kelas Good, meskipun jumlah instance lebih sedikit, model tetap menunjukkan hasil yang tinggi dengan presisi sebesar 0.94, recall sebesar 1.0, mAP@50 sebesar 0.983, dan mAP@50–95 sebesar 0.7.

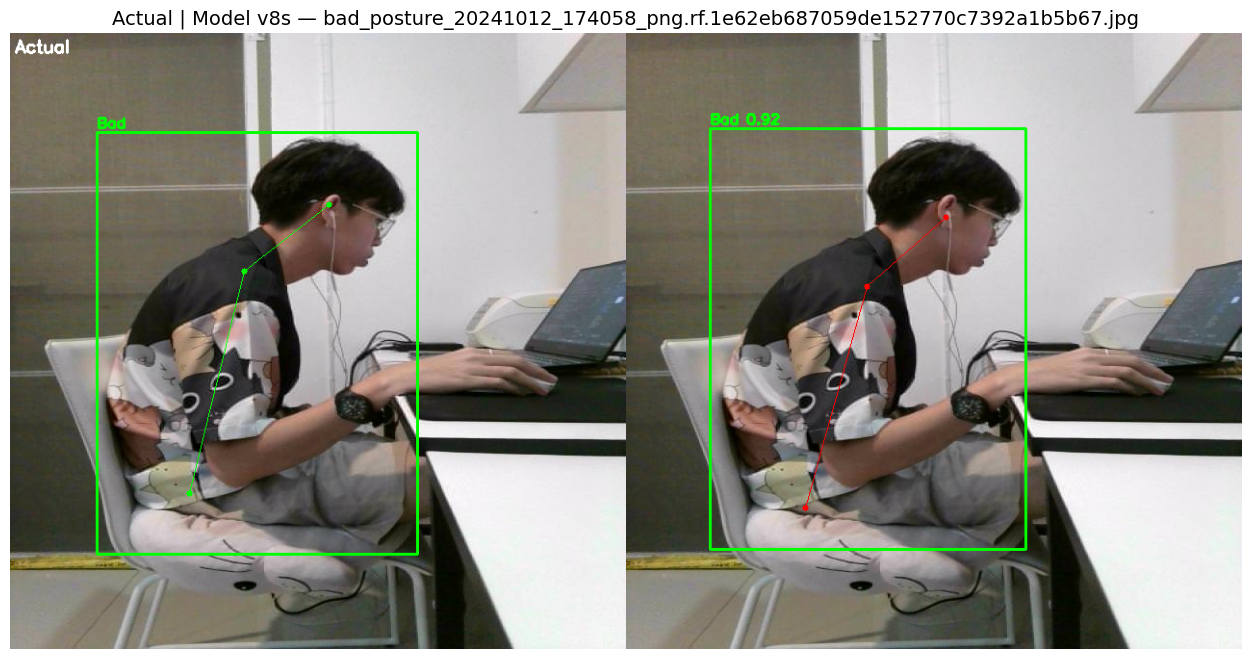
Dari segi efisiensi, waktu rata-rata yang dibutuhkan untuk memproses satu citra adalah 0.2 ms untuk preprocessing, 4.4 ms untuk inference, dan 2.1 ms untuk postprocessing. Seluruh hasil evaluasi disimpan pada direktori runs/pose/train2. Berdasarkan hasil tersebut, dapat disimpulkan bahwa model YOLOv8s-Pose yang dilatih mampu mendeteksi postur duduk dengan tingkat akurasi tinggi dan efisiensi waktu yang baik, sehingga layak digunakan dalam aplikasi pemantauan postur secara real-time maupun sistem pendukung ergonomi.

1. **Evaluasi data test**

Evaluasi model dengan keseluruhan data test (27) dengan membandingkan label aktual dengan hasil prediksi mendapatkan hasil yang bagus. Model bisa mendeteksi dengan nilai akurasi rata – rata diatas 80%. Namun, model juga cukup kesulitan dalam mendeteksi objek manusia apabila wajah menatap kamera. Selain itu, titik keypoint juga kesulitan mendeteksi letak pinggang dikarenakan posisi kamera yang sedikit keatas dan kebelakang. Berikut adalah contoh deteksi gambar test dengan hasil yang salah dan benar :







Apabila model tersebut digunakan dengan posisi kamera yang pasti maka hasil tersebut sudah cukup bagus. Namun, jika model digunakan dengan posisi kamera membelakangi atau didepan objek maka perlu ditambahkan beberapa dataset agar hasil deteksi bisa akurat.