Application of YOLOv5 Algorithm in Furniture Detection

Latar Belakang

Industri furniture merupakan salah satu sektor yang terus berkembang pesat seiring dengan meningkatnya kebutuhan konsumen akan produk-produk furniture yang berkualitas dan estetis. Salah satu tantangan utama dalam industri ini adalah manajemen inventaris dan penentuan kualitas produk yang memerlukan proses deteksi dan pengenalan furnitur yang cepat dan akurat. Oleh karena itu, teknologi deteksi objek berbasis visi komputer menjadi sangat relevan dalam mendukung operasional industri furnitur.

Deteksi objek adalah proses identifikasi dan klasifikasi objek dalam gambar atau video yang memiliki aplikasi luas di berbagai bidang, termasuk keamanan, otomotif, kesehatan, dan tentu saja industri furnitur. Dalam konteks industri furnitur, deteksi objek dapat digunakan untuk berbagai tujuan seperti pemantauan kualitas produk, manajemen inventaris otomatis, analisis tata letak ruangan, dan interaksi pelanggan yang lebih individual.

Salah satu algoritma deteksi objek yang menonjol dalam beberapa tahun terakhir adalah YOLO (You Simply See Once). YOLO dikenal karena kemampuannya melakukan deteksi objek secara real-time dengan kecepatan dan akurasi tinggi. Berbeda dengan metode deteksi objek lainnya yang sering kali memerlukan beberapa tahap proses, YOLO dapat melakukan deteksi objek dalam satu tahap, yang membuatnya lebih efisien.

YOLOv5 adalah versi terbaru dari keluarga algoritma YOLO yang memperkenalkan berbagai peningkatan signifikan dari versi sebelumnya. YOLOv5 menawarkan arsitektur yang lebih ringan dan lebih cepat dengan optimisasi yang memungkinkan deteksi multi-skala yang lebih baik. Hal ini menjadikan YOLOv5 sebagai pilihan yang sangat menjanjikan untuk aplikasi deteksi objek dalam industri yang memerlukan kecepatan dan efisiensi tinggi seperti industri furniture.

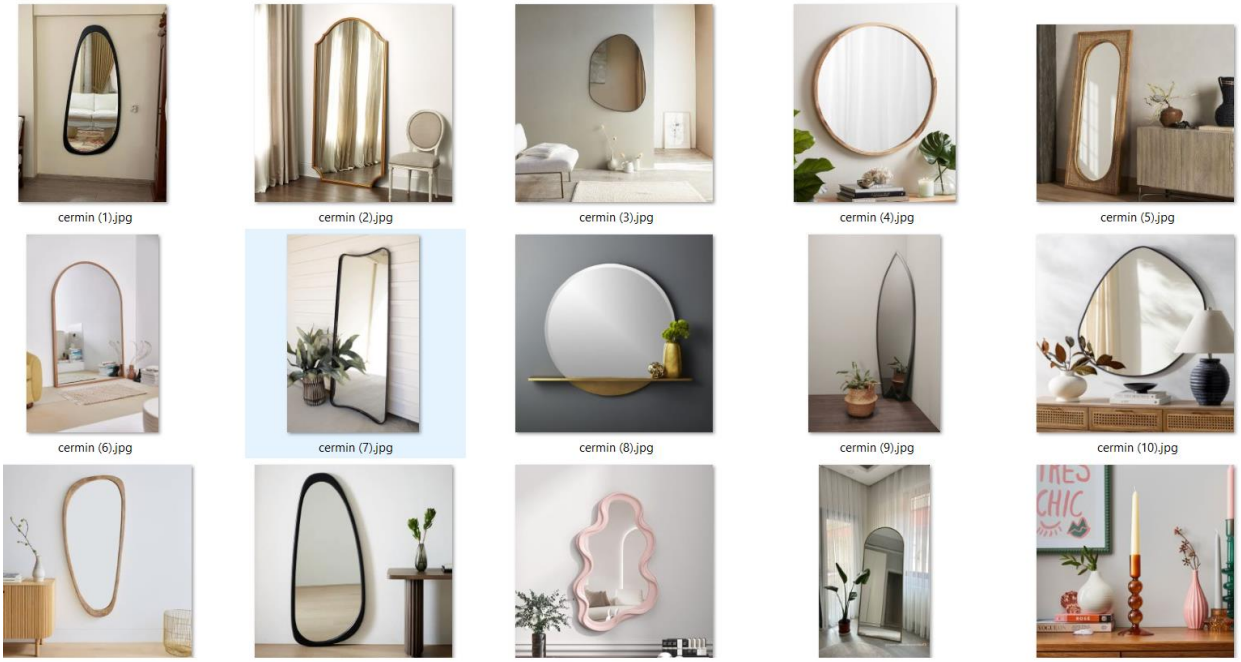
Penggunaan YOLOv5 dalam deteksi furnitur diharapkan dapat mengatasi beberapa keterbatasan dari metode konvensional. Metode sering kali memerlukan tenaga konvensional yang cukup banyak dan rentan terhadap kesalahan. Selain itu, deteksi manual cenderung memakan waktu dan biaya yang tidak sedikit. Dengan penerapan YOLOv5, proses deteksi furnitur dapat dilakukan secara otomatis dengan akurasi yang tinggi dan dalam waktu yang lebih singkat, sehingga meningkatkan efisiensi operasional dan mengurangi biaya produksi.

Penelitian ini bertujuan untuk menerapkan dan menyalakan kinerja algoritma YOLOv5 dalam mendeteksi berbagai jenis furnitur pada gambar. Evaluasi kinerja akan mencakup beberapa aspek penting seperti kecepatan inferensi, akurasi deteksi, serta perbandingan dengan algoritma deteksi objek lainnya seperti QuickerR-CNN dan SSD. Dengan demikian, diharapkan penelitian ini dapat memberikan kontribusi nyata bagi pengembangan teknologi deteksi objek dalam industri furnitur dan membuka peluang penerapan yang lebih luas di masa depan.

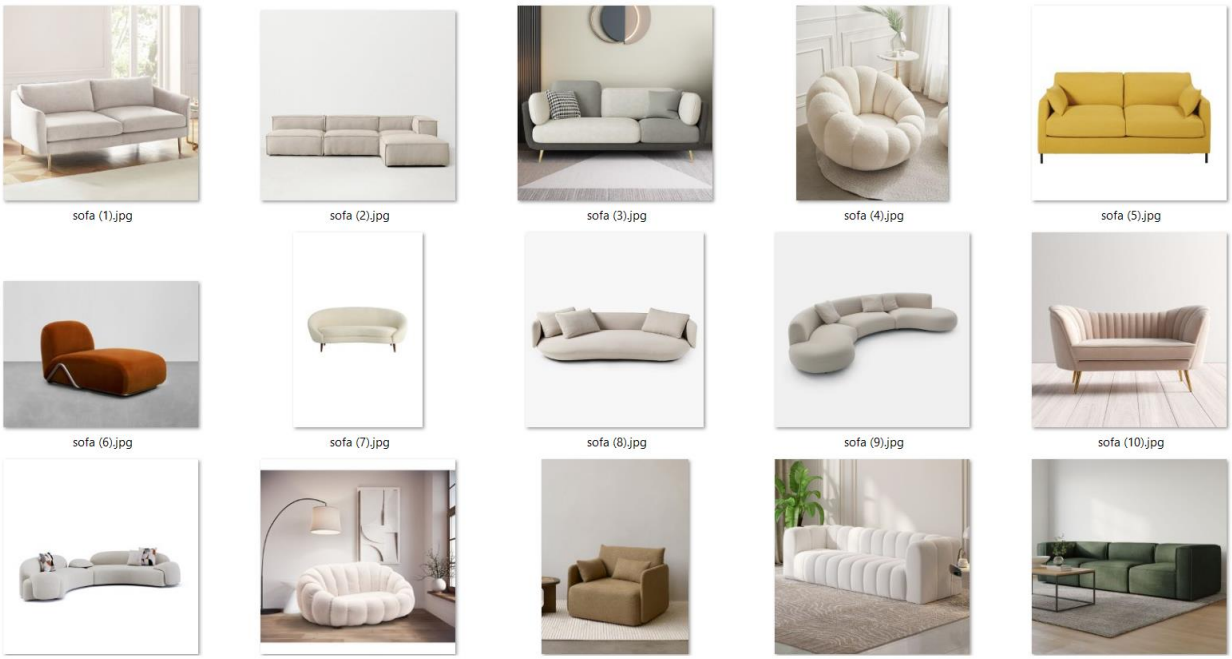
Melalui penelitian ini, diharapkan dapat menjawab pertanyaan-pertanyaan mendasar mengenai efektivitas dan efisiensi YOLOv5 dalam konteks deteksi furnitur. Penelitian ini juga diharapkan dapat menjadi referensi bagi para praktisi dan peneliti lainnya dalam mengembangkan solusi berbasis visi komputer yang lebih canggih dan terintegrasi untuk industri furnitur dan bidang terkait lainnya.

Data collection

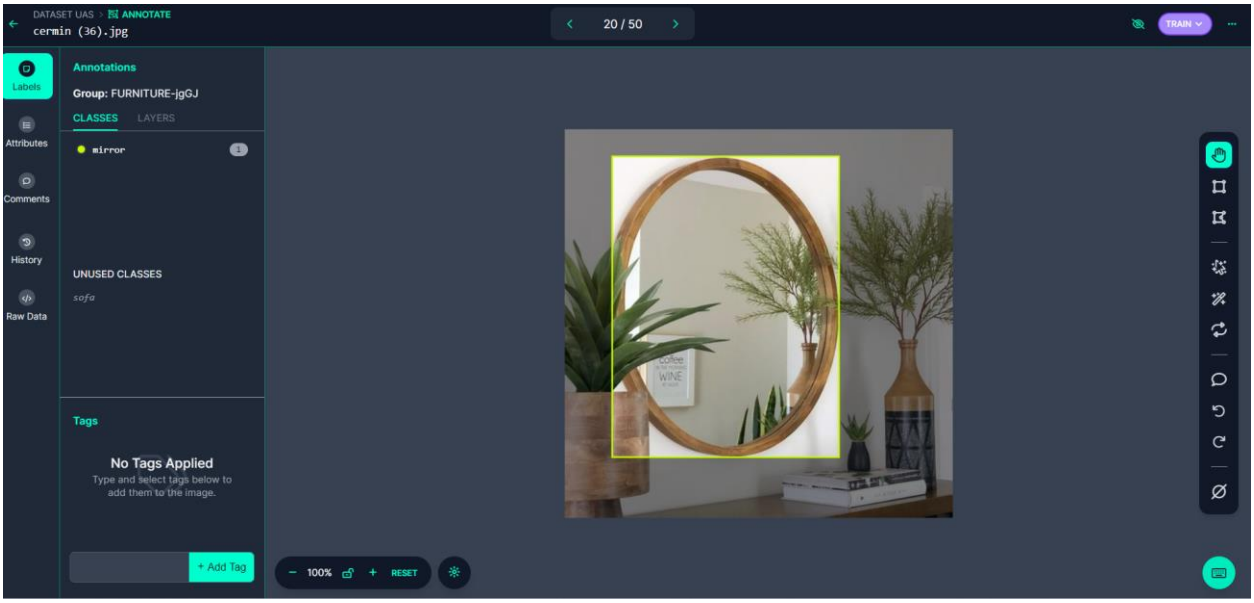
Dalam penelitian ini, proses pengumpulan dan anotasi dataset gambar dilakukan dengan menggunakan platform Pinterest dan Roboflow. Roboflow merupakan alat yang sangat efektif untuk anotasi gambar, memungkinkan peneliti untuk secara efisien menandai objek dalam gambar dengan bounding box dan label yang tepat. Tahapan pertama adalah pengumpulan gambar yang mengandung objek yang menjadi fokus penelitian, seperti mobil, sepeda, orang, anjing, dan kucing. Gambar-gambar ini dikumpulkan dari berbagai sumber, termasuk dataset publik dan internet, serta gambar yang diambil sendiri, memastikan variasi dalam kondisi pencahayaan, sudut pengambilan gambar, dan latar belakang untuk meningkatkan generalisasi model.Setelah gambar dikumpulkan, dilakukan pra-pemrosesan untuk mengubah ukuran, normalisasi, dan peningkatan kualitas gambar jika diperlukan. Selanjutnya, proses anotasi dilakukan menggunakan Roboflow. Roboflow menyediakan antarmuka yang intuitif untuk menandai setiap objek dalam gambar dengan bounding box dan label yang sesuai. Setiap anotasi diperiksa secara cermat untuk memastikan akurasi dan konsistensi. Selain itu, Roboflow mendukung verifikasi dan validasi anotasi dengan alat validasi otomatis untuk mendeteksi kesalahan.



Gambar 1 sample dataset



Gambar 2 sample dataset



Gambar 3 Annotate the images

Berikut kode program pengumpulan dan anotasi data

```
!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="xkxukZ8QTDpEcPHsfnFs")
project = rf.workspace("prita-hamidanti-dpbne").project("dataset-uas")
version = project.version(1)
dataset = version.download("yolov5")
```

Output

```
Collecting roboflow
  Downloading roboflow-1.1.36-py3-none-any.whl (76 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━
76.7/76.7 kB 696.1 kB/s eta 0:00:00
Requirement already satisfied: certifi in /usr/local/lib/python3.10/dist-
packages (from roboflow) (2024.7.4)
Collecting chardet==4.0.0 (from roboflow)
  Downloading chardet-4.0.0-py2.py3-none-any.whl (178 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━
178.7/178.7 kB 3.1 MB/s eta 0:00:00
Requirement already satisfied: idna==3.7 in /usr/local/lib/python3.10/dist-
packages (from roboflow) (3.7)
Requirement already satisfied: cycler in /usr/local/lib/python3.10/dist-
packages (from roboflow) (0.12.1)
Requirement already satisfied: kiwisolver>=1.3.1 in
/usr/local/lib/python3.10/dist-packages (from roboflow) (1.4.5)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-
packages (from roboflow) (3.7.1)
Requirement already satisfied: numpy>=1.18.5 in /usr/local/lib/python3.10/dist-
packages (from roboflow) (1.25.2)
Requirement already satisfied: opencv-python-headless==4.10.0.84 in
/usr/local/lib/python3.10/dist-packages (from roboflow) (4.10.0.84)
Requirement already satisfied: Pillow>=7.1.2 in /usr/local/lib/python3.10/dist-
packages (from roboflow) (9.4.0)
Requirement already satisfied: python-dateutil in
/usr/local/lib/python3.10/dist-packages (from roboflow) (2.8.2)
Collecting python-dotenv (from roboflow)
  Downloading python_dotenv-1.0.1-py3-none-any.whl (19 kB)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-
packages (from roboflow) (2.31.0)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages
(from roboflow) (1.16.0)
Requirement already satisfied: urllib3>=1.26.6 in
/usr/local/lib/python3.10/dist-packages (from roboflow) (2.0.7)
Requirement already satisfied: tqdm>=4.41.0 in /usr/local/lib/python3.10/dist-
packages (from roboflow) (4.66.4)
Requirement already satisfied: PyYAML>=5.3.1 in /usr/local/lib/python3.10/dist-
packages (from roboflow) (6.0.1)
Collecting requests-toolbelt (from roboflow)
  Downloading requests_toolbelt-1.0.0-py2.py3-none-any.whl (54 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━
54.5/54.5 kB 2.9 MB/s eta 0:00:00
Collecting filetype (from roboflow)
  Downloading filetype-1.2.0-py2.py3-none-any.whl (19 kB)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->roboflow) (1.2.1)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->roboflow) (4.53.1)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->roboflow) (24.1)
```

```
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->roboflow) (3.1.2)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests->roboflow) (3.3.2)
Installing collected packages: filetype, python-dotenv, chardet, requests-
toolbelt, roboflow
  Attempting uninstall: chardet
    Found existing installation: chardet 5.2.0
    Uninstalling chardet-5.2.0:
      Successfully uninstalled chardet-5.2.0
Successfully installed chardet-4.0.0 filetype-1.2.0 python-dotenv-1.0.1
requests-toolbelt-1.0.0 roboflow-1.1.36
loading Roboflow workspace...
loading Roboflow project...
Downloading Dataset Version Zip in dataset-uas-1 to yolov5pytorch::
100%|██████████| 3619/3619 [00:00<00:00, 53639.92it/s]

Extracting Dataset Version Zip to dataset-uas-1 in yolov5pytorch::
100%|██████████| 224/224 [00:00<00:00, 5745.90it/s]
```

Model Training

Mengkloning Repositori YOLOv5 dan Menginstal Dependensi:

```
#clone YOLOv5 and
!git clone https://github.com/ultralytics/yolov5  # clone repo
%cd yolov5
%pip install -qr requirements.txt # install dependencies
%pip install -q roboflow

import torch
import os
from IPython.display import Image, clear_output  # to display images

print(f"Setup complete. Using torch {torch.__version__}
({torch.cuda.get_device_properties(0).name if torch.cuda.is_available() else
'CPU'})")
```

Output

```
Cloning into 'yolov5'...
remote: Enumerating objects: 16802, done.
remote: Counting objects: 100% (123/123), done.
remote: Compressing objects: 100% (96/96), done.
remote: Total 16802 (delta 54), reused 67 (delta 27), pack-reused 16679
Receiving objects: 100% (16802/16802), 15.47 MiB | 12.08 MiB/s, done.
Resolving deltas: 100% (11533/11533), done.
/content/yolov5
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━
207.3/207.3 kB 1.6 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━
4.5/4.5 MB 22.1 MB/s eta 0:00:00
```

```
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━
64.9/64.9 kB 6.9 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━
802.9/802.9 kB 30.8 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━
2.3/2.3 MB 52.8 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━
62.7/62.7 kB 6.3 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━
21.3/21.3 MB 44.4 MB/s eta 0:00:00
```

Selanjutnya mendownload dataset dari Roboflow dari key API yang telah didapatkan

```
# set up environment
os.environ["DATASET_DIRECTORY"] = "/content/DATA_UAS-1"
from roboflow import Roboflow
rf    =    Roboflow(api_key="xkxukZ8QTDpEcPHsfnFs",    model_format="yolov5",
notebook="ultralytics")
dataset       =       rf.workspace("prita-hamidanti-dpbne").project("dataset-
uas").version("1").download(location="/content/my-datasets")
```

output

```
loading Roboflow workspace...
loading Roboflow project...
Downloading Dataset Version Zip in /content/my-datasets to yolov5pytorch::
100%|██████████| 3619/3619 [00:00<00:00, 52276.23it/s]


Extracting Dataset Version Zip to /content/my-datasets in yolov5pytorch::
100%|██████████| 224/224 [00:00<00:00, 4922.27it/s]
```

Model Evaluation

```
!python    train.py    --img    416    --batch    16    --epochs    25    --data
{dataset.location}/data.yaml --weights yolov5s.pt --cache
```

Berdasarkan kode program dia atas train.py dari YOLOv5 untuk melatih model. Mengatur parameter pelatihan seperti ukuran gambar (--img 416), ukuran batch (--batch 16), jumlah epoch (--epochs 25), file konfigurasi data (--data {dataset.location}/data.yaml), dan bobot awal (--weights yolov5s.pt). Menggunakan opsi --cache untuk caching dataset ke dalam RAM.

Output

```
2024-07-20 04:22:11.894479: E
external/local_xla/xla/stream_executor/cuda/cuda_dnn.cc:9261] Unable to
register cuDNN factory: Attempting to register factory for plugin cuDNN when
one has already been registered
2024-07-20 04:22:11.894555: E
external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:607] Unable to
register cuFFT factory: Attempting to register factory for plugin cuFFT when
one has already been registered
```

2024-07-20 04:22:11.896586: E external/local_xla/xla/stream_executor/cuda/cuda_blas.cc:1515] Unable to register cuBLAS factory: Attempting to register factory for plugin cuBLAS when one has already been registered
train: weights=yolov5s.pt, cfg=, data=/content/my-datasets/data.yaml, hyp=data/hyps/hyp.scratch-low.yaml, epochs=25, batch_size=16, imgsz=416, rect=False, resume=False, nosave=False, noval=False, noautoanchor=False, noplots=False, evolve=None, evolve_population=data/hyps, resume_evolve=None, bucket=, cache=ram, image_weights=False, device=, multi_scale=False, single_cls=False, optimizer=SGD, sync_bn=False, workers=8, project=runs/train, name=exp, exist_ok=False, quad=False, cos_lr=False, label_smoothing=0.0, patience=100, freeze=[0], save_period=-1, seed=0, local_rank=-1, entity=None, upload_dataset=False, bbox_interval=-1, artifact_alias=latest, ndjson_console=False, ndjson_file=False
github: up to date with https://github.com/ultralytics/yolov5 ✅
YOLOv5 🚀 v7.0-345-g8003649c Python-3.10.12 torch-2.3.1+cu121 CPU

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, obj=1.0, obj_pw=1.0, iou_t=0.2, anchor_t=4.0, fl_gamma=0.0, hsv_h=0.015, hsv_s=0.7, hsv_v=0.4, degrees=0.0, translate=0.1, scale=0.5, shear=0.0, perspective=0.0, flipud=0.0, fliplr=0.5, mosaic=1.0, mixup=0.0, copy_paste=0.0
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/
Overriding model.yaml nc=80 with nc=2

	from	n	params	module	arguments
0	-1	1	3520	models.common.Conv	[3, 32, 6, 2, 2]
1	-1	1	18560	models.common.Conv	[32, 64, 3, 2]
2	-1	1	18816	models.common.C3	[64, 64, 1]
3	-1	1	73984	models.common.Conv	[64, 128, 3, 2]
4	-1	2	115712	models.common.C3	[128, 128, 2]
5	-1	1	295424	models.common.Conv	[128, 256, 3, 2]
6	-1	3	625152	models.common.C3	[256, 256, 3]
7	-1	1	1180672	models.common.Conv	[256, 512, 3, 2]
8	-1	1	1182720	models.common.C3	[512, 512, 1]
9	-1	1	656896	models.common.SPPF	[512, 512, 5]

```
 10                -1  1    131584  models.common.Conv
[512, 256, 1, 1]
 11                -1  1         0  torch.nn.modules.upsampling.Upsample
[None, 2, 'nearest']
 12           [-1, 6]  1         0  models.common.Concat
[1]
 13                -1  1    361984  models.common.C3
[512, 256, 1, False]
 14                -1  1     33024  models.common.Conv
[256, 128, 1, 1]
 15                -1  1         0  torch.nn.modules.upsampling.Upsample
[None, 2, 'nearest']
 16           [-1, 4]  1         0  models.common.Concat
[1]
 17                -1  1     90880  models.common.C3
[256, 128, 1, False]
 18                -1  1    147712  models.common.Conv
[128, 128, 3, 2]
 19          [-1, 14]  1         0  models.common.Concat
[1]
 20                -1  1    296448  models.common.C3
[256, 256, 1, False]
 21                -1  1    590336  models.common.Conv
[256, 256, 3, 2]
 22          [-1, 10]  1         0  models.common.Concat
[1]
 23                -1  1   1182720  models.common.C3
[512, 512, 1, False]
 24      [17, 20, 23]  1     18879  models.yolo.Detect
[2, [[10, 13, 16, 30, 33, 23], [30, 61, 62, 45, 59, 119], [116, 90, 156, 198,
373, 326]], [128, 256, 512]]
Model summary: 214 layers, 7025023 parameters, 7025023 gradients, 16.0 GFLOPs

Transferred 343/349 items from yolov5s.pt
optimizer: SGD(lr=0.01) with parameter groups 57 weight(decay=0.0), 60
weight(decay=0.0005), 60 bias
albumentations: Blur(p=0.01, blur_limit=(3, 7)), MedianBlur(p=0.01,
blur_limit=(3, 7)), ToGray(p=0.01), CLAHE(p=0.01, clip_limit=(1, 4.0),
tile_grid_size=(8, 8))
train: Scanning /content/my-datasets/train/labels.cache... 85 images, 1
backgrounds, 0 corrupt: 100% 85/85 [00:00<?, ?it/s]
train: Caching images (0.0GB ram): 100% 85/85 [00:00<00:00, 161.03it/s]
/usr/lib/python3.10/multiprocessing/popen_fork.py:66: RuntimeWarning:
os.fork() was called. os.fork() is incompatible with multithreaded code, and
JAX is multithreaded, so this will likely lead to a deadlock.
  self.pid = os.fork()
val: Scanning /content/my-datasets/valid/labels.cache... 10 images, 0
backgrounds, 0 corrupt: 100% 10/10 [00:00<?, ?it/s]
val: Caching images (0.0GB ram): 100% 10/10 [00:00<00:00, 31.51it/s]

AutoAnchor: 3.65 anchors/target, 1.000 Best Possible Recall (BPR). Current
anchors are a good fit to dataset ✅
```

```
Plotting labels to runs/train/exp2/labels.jpg...
Image sizes 416 train, 416 val
Using 2 dataloader workers
Logging results to runs/train/exp2
Starting training for 25 epochs...

      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
       0/24         0G     0.1152    0.01862    0.02917         12
416: 100% 6/6 [01:28<00:00, 14.68s/it]
                 Class     Images  Instances          P          R      mAP50
mAP50-95:   0% 0/1 [00:00<?, ?it/s]WARNING ⚠️ NMS time limit 1.000s exceeded
                 Class     Images  Instances          P          R      mAP50
mAP50-95: 100% 1/1 [00:04<00:00,  4.46s/it]
                   all         10         11    0.00421      0.583      0.116
0.0139

      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
       1/24         0G     0.1116    0.02096    0.02932         18
416: 100% 6/6 [01:10<00:00, 11.81s/it]
                 Class     Images  Instances          P          R      mAP50
mAP50-95:   0% 0/1 [00:00<?, ?it/s]WARNING ⚠️ NMS time limit 1.000s exceeded
                 Class     Images  Instances          P          R      mAP50
mAP50-95: 100% 1/1 [00:04<00:00,  4.46s/it]
                   all         10         11    0.00407      0.583      0.119
0.0162

      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
       2/24         0G    0.09591    0.02435    0.02928         15
416: 100% 6/6 [01:04<00:00, 10.74s/it]
                 Class     Images  Instances          P          R      mAP50
mAP50-95: 100% 1/1 [00:03<00:00,  3.79s/it]
                   all         10         11    0.00366          1      0.132
0.0387

      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
       3/24         0G    0.08691     0.0247    0.02628         11
416: 100% 6/6 [01:11<00:00, 11.99s/it]
                 Class     Images  Instances          P          R      mAP50
mAP50-95: 100% 1/1 [00:03<00:00,  3.85s/it]
                   all         10         11     0.0037          1     0.0537
0.0174

      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
       4/24         0G    0.08392    0.02634    0.02736         19
416: 100% 6/6 [01:10<00:00, 11.81s/it]
                 Class     Images  Instances          P          R      mAP50
mAP50-95:   0% 0/1 [00:00<?, ?it/s]WARNING ⚠️ NMS time limit 1.000s exceeded
                 Class     Images  Instances          P          R      mAP50
mAP50-95: 100% 1/1 [00:04<00:00,  4.55s/it]
                   all         10         11       0.12      0.367      0.192
0.0306
```

```
      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
       5/24         0G    0.07366    0.02417    0.02321         13
416: 100% 6/6 [01:02<00:00, 10.40s/it]
                 Class     Images  Instances          P          R      mAP50
mAP50-95: 100% 1/1 [00:04<00:00,  4.03s/it]
                   all         10         11      0.321      0.283      0.298
0.0772

      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
       6/24         0G    0.07358    0.02641    0.02341         17
416: 100% 6/6 [01:11<00:00, 11.90s/it]
                 Class     Images  Instances          P          R      mAP50
mAP50-95: 100% 1/1 [00:03<00:00,  3.77s/it]
                   all         10         11      0.472      0.467      0.409
0.176

      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
       7/24         0G     0.0652    0.02626    0.02135         10
416: 100% 6/6 [01:11<00:00, 11.90s/it]
                 Class     Images  Instances          P          R      mAP50
mAP50-95: 100% 1/1 [00:03<00:00,  3.97s/it]
                   all         10         11      0.745      0.467      0.618
0.313

      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
       8/24         0G    0.06573    0.02231     0.0206         12
416: 100% 6/6 [01:04<00:00, 10.82s/it]
                 Class     Images  Instances          P          R      mAP50
mAP50-95: 100% 1/1 [00:03<00:00,  3.09s/it]
                   all         10         11      0.734      0.524      0.647
0.281

      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
       9/24         0G    0.06107    0.02387    0.01731         14
416: 100% 6/6 [01:20<00:00, 13.34s/it]
                 Class     Images  Instances          P          R      mAP50
mAP50-95: 100% 1/1 [00:02<00:00,  2.72s/it]
                   all         10         11       0.83      0.697      0.856
0.462

      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
      10/24         0G    0.06173    0.02452    0.01732         18
416: 100% 6/6 [01:13<00:00, 12.22s/it]
                 Class     Images  Instances          P          R      mAP50
mAP50-95: 100% 1/1 [00:03<00:00,  3.77s/it]
                   all         10         11      0.519      0.681      0.768
0.397

      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
      11/24         0G     0.0555    0.02313    0.01598         12
416: 100% 6/6 [01:04<00:00, 10.70s/it]
                 Class     Images  Instances          P          R      mAP50
mAP50-95: 100% 1/1 [00:03<00:00,  3.42s/it]
```

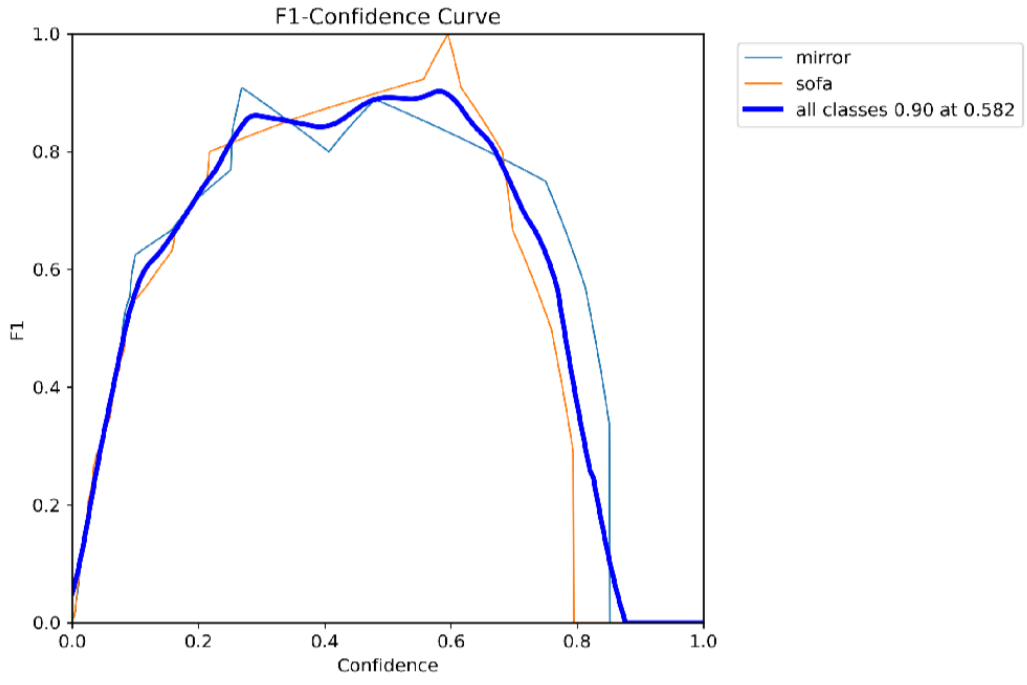
```
                 all         10         11      0.817      0.781      0.887
0.474

      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
      12/24         0G    0.05979    0.02099    0.01364         14
416: 100% 6/6 [01:12<00:00, 12.16s/it]
                 Class     Images  Instances          P          R      mAP50
mAP50-95: 100% 1/1 [00:03<00:00,  3.53s/it]
                 all         10         11      0.678      0.865      0.869
0.421

      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
      13/24         0G    0.05367    0.02118    0.01491         14
416: 100% 6/6 [01:11<00:00, 12.00s/it]
                 Class     Images  Instances          P          R      mAP50
mAP50-95: 100% 1/1 [00:03<00:00,  3.38s/it]
                 all         10         11      0.527      0.999      0.813
0.317

      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
      14/24         0G    0.05831    0.02116    0.01195         18
416: 100% 6/6 [01:03<00:00, 10.60s/it]
                 Class     Images  Instances          P          R      mAP50
mAP50-95: 100% 1/1 [00:03<00:00,  3.49s/it]
                 all         10         11      0.778      0.663      0.842
0.597

      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
      15/24         0G    0.04912    0.02067    0.01363         15
416: 100% 6/6 [01:10<00:00, 11.76s/it]
                 Class     Images  Instances          P          R      mAP50
mAP50-95: 100% 1/1 [00:03<00:00,  3.46s/it]
                 all         10         11      0.568      0.872      0.796
0.444

      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
      16/24         0G    0.04949     0.0173    0.01144         12
416: 100% 6/6 [01:10<00:00, 11.77s/it]
                 Class     Images  Instances          P          R      mAP50
mAP50-95: 100% 1/1 [00:03<00:00,  3.45s/it]
                 all         10         11      0.828      0.787       0.92
0.632

      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
      17/24         0G    0.04949    0.01944    0.01159         14
416: 100% 6/6 [01:07<00:00, 11.18s/it]
                 Class     Images  Instances          P          R      mAP50
mAP50-95: 100% 1/1 [00:03<00:00,  3.38s/it]
                 all         10         11      0.744      0.717      0.835
0.469

      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
```

```
      18/24         0G    0.04954    0.01909    0.01061         15
416: 100% 6/6 [01:10<00:00, 11.71s/it]
                 Class     Images  Instances          P          R      mAP50
mAP50-95: 100% 1/1 [00:02<00:00,  2.68s/it]
                   all         10         11       0.78       0.77      0.889
0.591

      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
      19/24         0G    0.05023    0.02084   0.009693         21
416: 100% 6/6 [01:09<00:00, 11.58s/it]
                 Class     Images  Instances          P          R      mAP50
mAP50-95: 100% 1/1 [00:03<00:00,  3.43s/it]
                   all         10         11      0.765      0.594      0.797
0.471

      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
      20/24         0G    0.04504    0.01801   0.008939         11
416: 100% 6/6 [01:05<00:00, 10.98s/it]
                 Class     Images  Instances          P          R      mAP50
mAP50-95: 100% 1/1 [00:02<00:00,  2.85s/it]
                   all         10         11      0.907      0.717      0.911
0.585

      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
      21/24         0G    0.04325    0.01872   0.008819         12
416: 100% 6/6 [01:12<00:00, 12.09s/it]
                 Class     Images  Instances          P          R      mAP50
mAP50-95: 100% 1/1 [00:02<00:00,  2.07s/it]
                   all         10         11      0.826      0.855      0.938
0.585

      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
      22/24         0G    0.03969    0.01778   0.006573          9
416: 100% 6/6 [01:06<00:00, 11.05s/it]
                 Class     Images  Instances          P          R      mAP50
mAP50-95: 100% 1/1 [00:02<00:00,  2.66s/it]
                   all         10         11      0.855      0.938      0.975
0.658

      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
      23/24         0G    0.03936    0.01689   0.007576         13
416: 100% 6/6 [01:12<00:00, 12.05s/it]
                 Class     Images  Instances          P          R      mAP50
mAP50-95: 100% 1/1 [00:02<00:00,  2.34s/it]
                   all         10         11      0.786        0.9      0.938
0.684

      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
      24/24         0G    0.04087    0.01864   0.007506         17
416: 100% 6/6 [01:12<00:00, 12.15s/it]
                 Class     Images  Instances          P          R      mAP50
mAP50-95: 100% 1/1 [00:02<00:00,  2.10s/it]
```

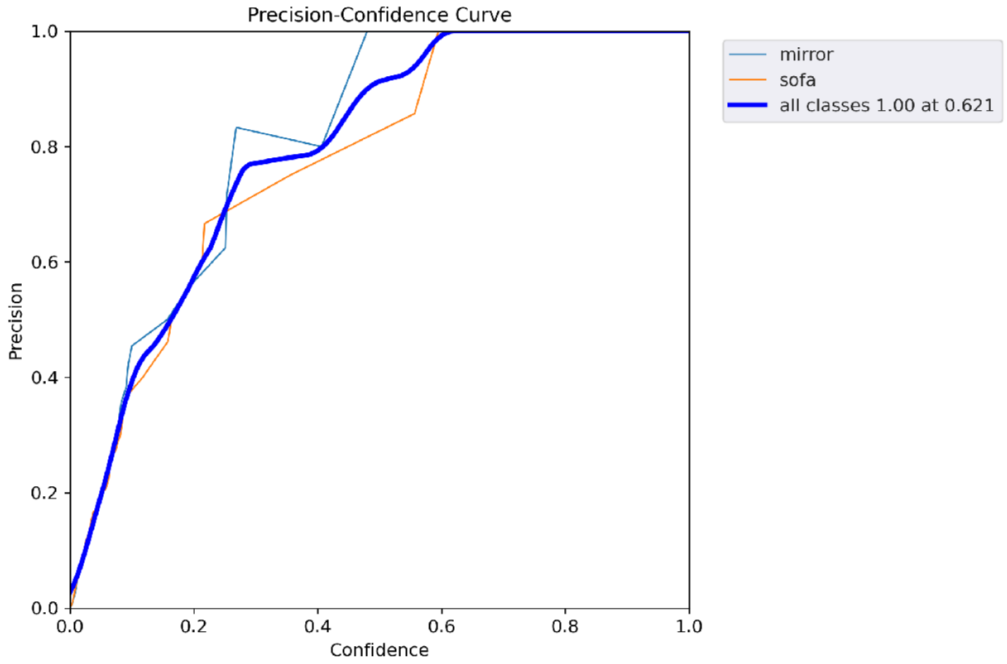
```
                   all         10         11      0.928      0.872      0.978
0.705

25 epochs completed in 0.517 hours.
Optimizer stripped from runs/train/exp2/weights/last.pt, 14.3MB
Optimizer stripped from runs/train/exp2/weights/best.pt, 14.3MB

Validating runs/train/exp2/weights/best.pt...
Fusing layers...
Model summary: 157 layers, 7015519 parameters, 0 gradients, 15.8 GFLOPs
                 Class     Images  Instances          P          R      mAP50
mAP50-95: 100% 1/1 [00:03<00:00,  3.26s/it]
                   all         10         11      0.928      0.872      0.978
0.697
                mirror         10          5          1      0.744      0.962
0.729
                  sofa         10          6      0.857          1      0.995
0.666
Results saved to runs/train/exp2
```

Berikut merupakan metrics evaluasi untuk mengevaluasi model yang telah dibuat.

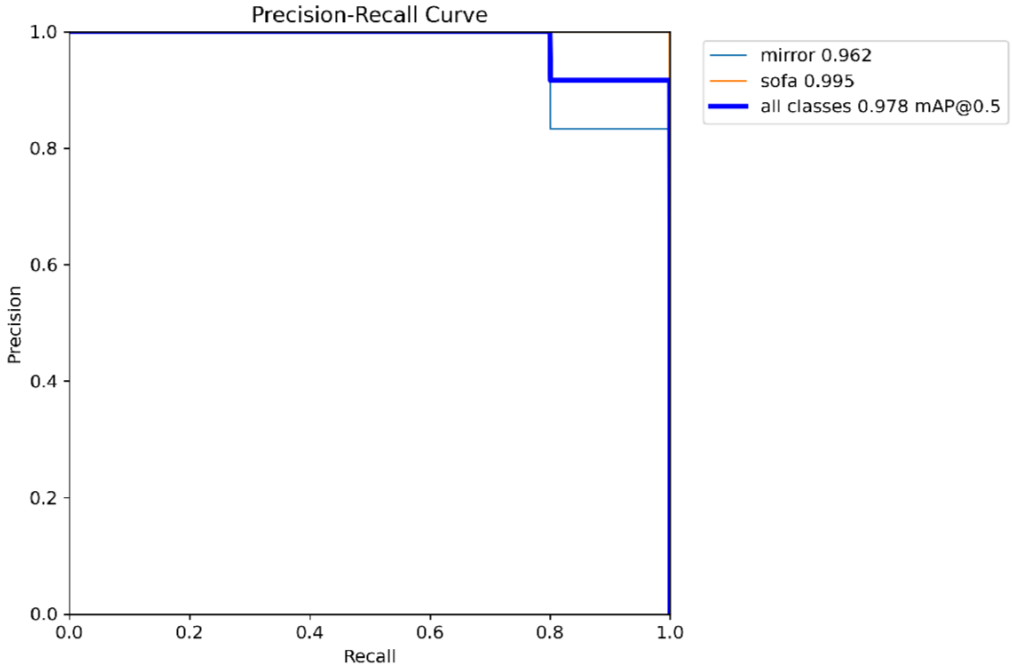
```
# Start tensorboard
# Launch after you have started training
# logs save in the folder "runs"
%load_ext tensorboard
%tensorboard --logdir runs
```



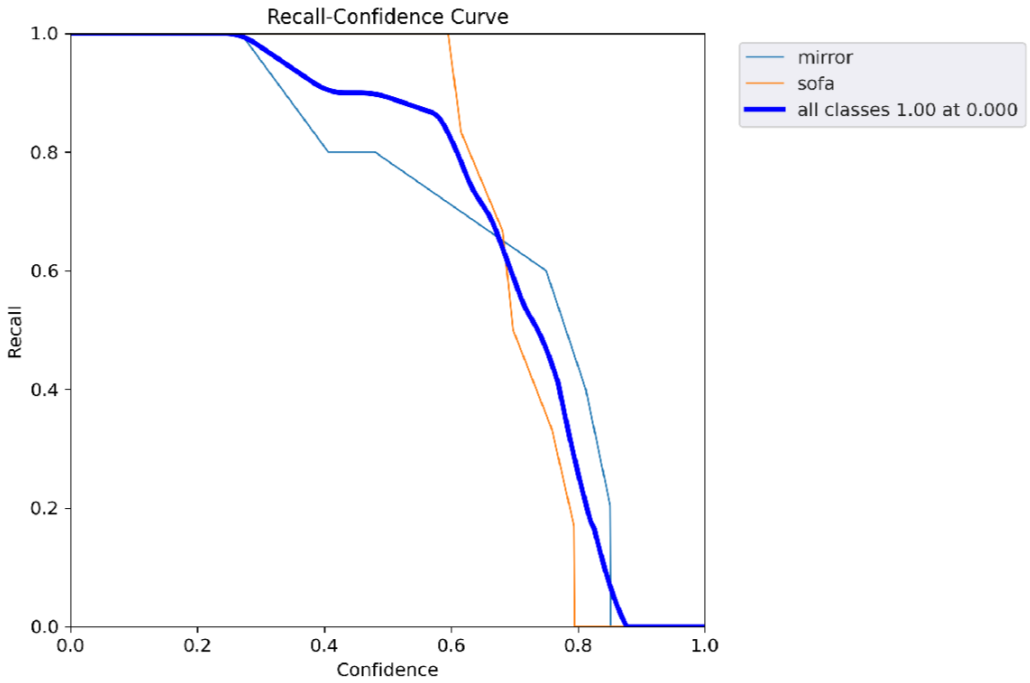
Gambar 4 F1 Score



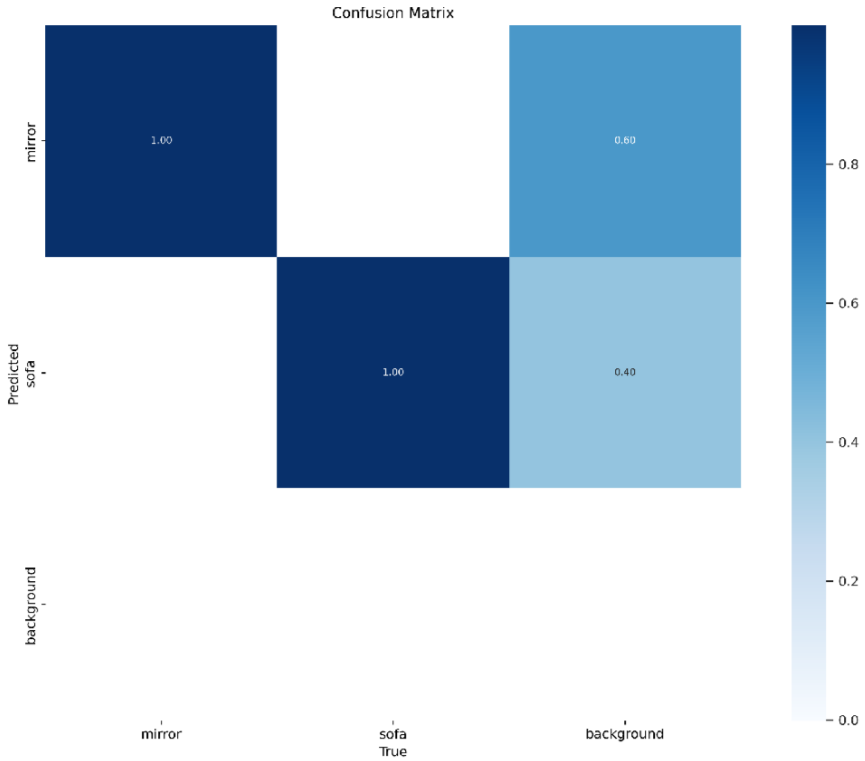
Gambar 5 Precision-Confidence



Gambar 6 Precision-Recall



Gambar 7 Recall-Confidence



Gambar 8 Confusion Matrix

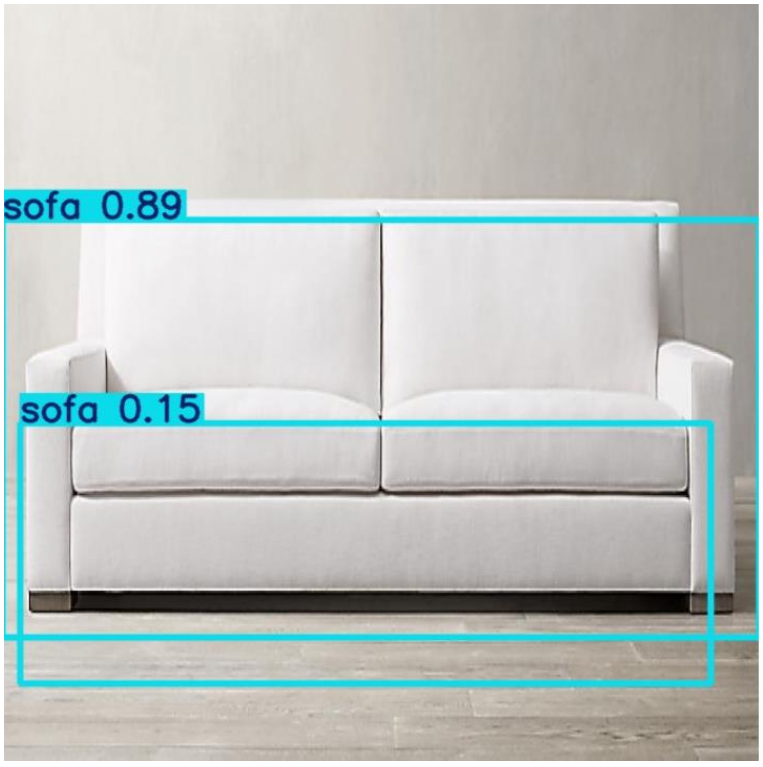
Mendeteksi menggunakan model terlatih

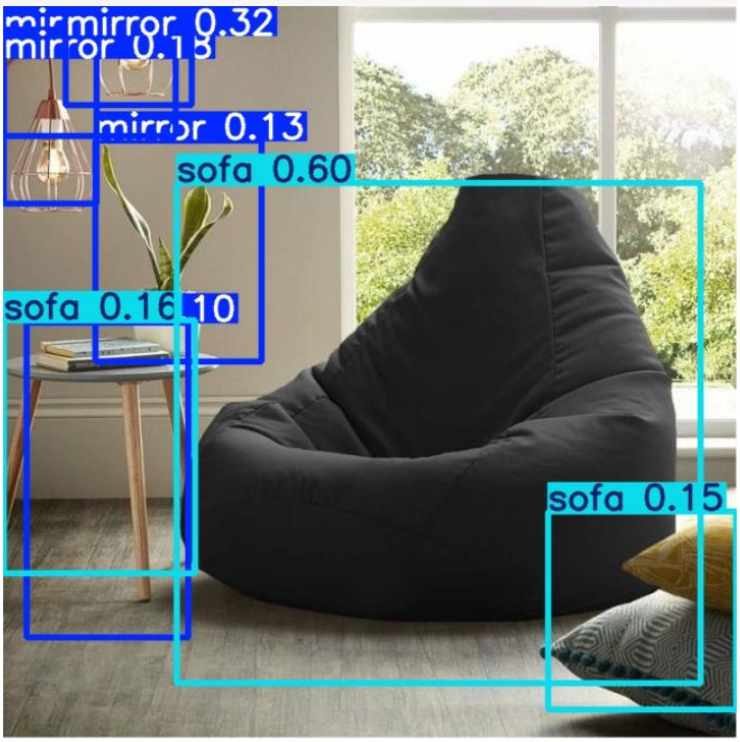
```
!python detect.py --weights runs/train/exp2/weights/best.pt --img 416 --conf
0.1 --source {dataset.location}/test/images
```

Berdasarkan kode program di atas skrip detect.py dari YOLOv5 untuk mendeteksi objek pada gambar uji. Mengatur parameter seperti bobot model (--weights runs/train/exp/weights/best.pt), ukuran gambar (--img 416), ambang batas kepercayaan (--conf 0.1), dan sumber gambar (--source {dataset.location}/test/images).

Visual example of model detection on test image.

```
#display inference on ALL test images

import glob

from IPython.display import Image, display

i=0

for imageName in glob.glob('/content/yolov5/runs/detect/exp/*.jpg'):

  i +=1

  if i<13:

    display(Image(filename=imageName))

print("\n")
```





Gambar yang ditampilkan adalah hasil dari deteksi objek menggunakan model YOLOv5. Kotak deteksi dan label menunjukkan bahwa model berhasil mengenali beberapa objek dalam gambar, seperti "sofa" dan "mirror" dengan tingkat kepercayaan yang bervariasi. Proses ini melibatkan menjalankan skrip deteksi dan menampilkan hasilnya dalam notebook.

Gambar yang ditampilkan adalah hasil dari deteksi objek menggunakan model YOLOv5. Kotak deteksi dan label menunjukkan bahwa model berhasil mengenali objek dalam gambar sebagai

"sofa" dengan tingkat kepercayaan 88%. Proses ini melibatkan menjalankan skrip deteksi dan menampilkan hasilnya dalam notebook.