DAFTAR PUSTAKA

Anwar, H., 2022. Sistem Deteksi dan Estimasi Jarak Tangga secara *Real-time* berbasis Android untuk Penyandang Tunanetra.

Brebahama, A., Listyandini, R.A., Fakultas Psikologi Universitas YARSI, 2016. Gambaran Tingkat Kesejahteraan Psikologis Penyandang Tunanetra Dewasa Muda. mps 02, 1–10. https://doi.org/10.21776/ub.mps.2016.002.01.1

Hanny Hafiar, Yanti Setianti, Priyo Subekti, Anwar Sani, 2020. Blind Code pada Uang Kertas Rupiah Pesan Komunikasi dan Komunikasi Pesan kepada Publik Disabilitas Netra. Jurnal Kawistara 10, 328–342. https://doi.org/ 10.22146/kawistara. 48865

Ioannis Markoulidakis, Ioannis Rallis, Ioannis Georgoulas, George Kopsiaftis, Anastasios Doulamis, Nikolaos Doulamis. 2021. Multiclass Confusion Matrix Reducion Method and Its Application on Net Promoter Score Classification Problem. https://doi.org/10.3390/technologies9040081

Kevin Maulana Azhar, Imam Santoso, Yosua Alvin Adi Soetrisno, 2021. Implementasi Deep Learning Menggunakan Metode Convolutional Neural Network dan Algoritma YOLO dalam Sistem Pendeteksi Uang Kertas Rupiah Bagi Penyandang Low Vision. Transient 10, 502–510.

Krishna, C.R., n.d. Computer Vision:Foundations and Applications.

Mahesh Pawar, Ketan Desai, Sanjay Gupta, Trupti Chavan, Kirti Mhamunkar, 2019. Object Detection and Currency Recognition using CNN. Cikitusi Journal for Multidisciplinary Reasearch 6, 357–362.

Priyal Doshi, 2019. Currency Feature Extraction using Image Processing Techniques.

Putri, R.Z., 2023. Klasifikasi Tanaman Obat menggunakan *Multiclass Support Vector Machine* berbasis Android.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You Only Look Once: Unified, Real-Time Object Detection, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Presented at the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Las Vegas, NV, USA, pp. 779–788. https://doi.org/10.1109/CVPR.2016.91

Utomo & Nadya Muniroh. 2020. Keterampilan Orientasi Mobilitas (OM) bagi Tunanetra. Nizamiah Learning Center.

Wahyu Pradika Candrawasih, 2017. Rancang Bangun Alat Bantu Pendeteksi Nominal Uang Kertas untuk Tunanetra Menggunakan Kamera Berbasis Raspberry Pi.

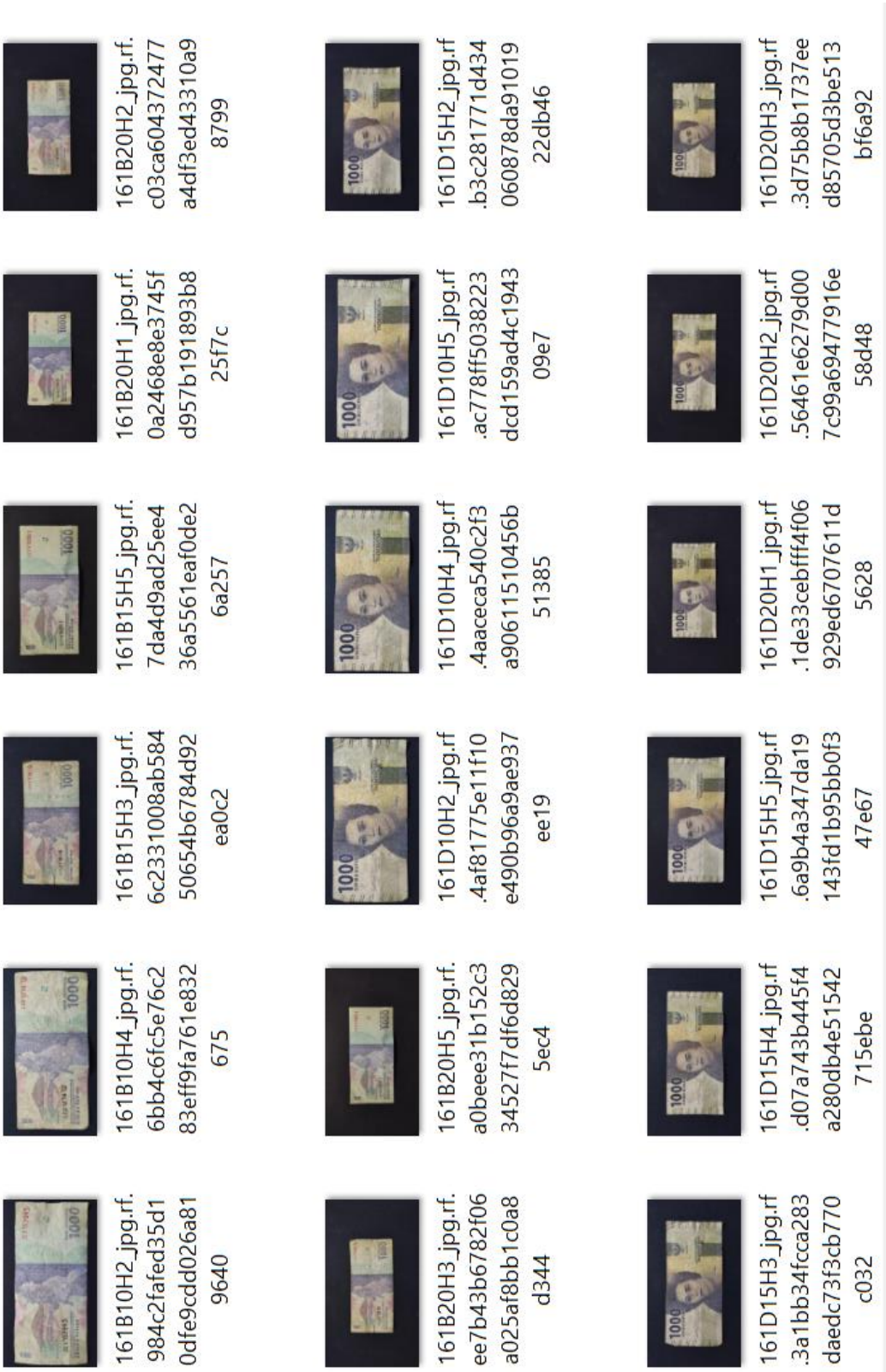
Wanda Hamidah, Nurul Amanda Pratiwi Hasbullah, Tsabita Syalza Billa Irawan, Andi Baso Kaswar, 2022. Deteksi Nominal Uang Kertas Menggunakan OCR (Optical Character Recognition). Techno Xplore Jurnal Ilmu Komputer dan Teknologi Informasi 7.

Yoshikai Shiari, 1987. Three-Dimensional Computer Vision. Japan.

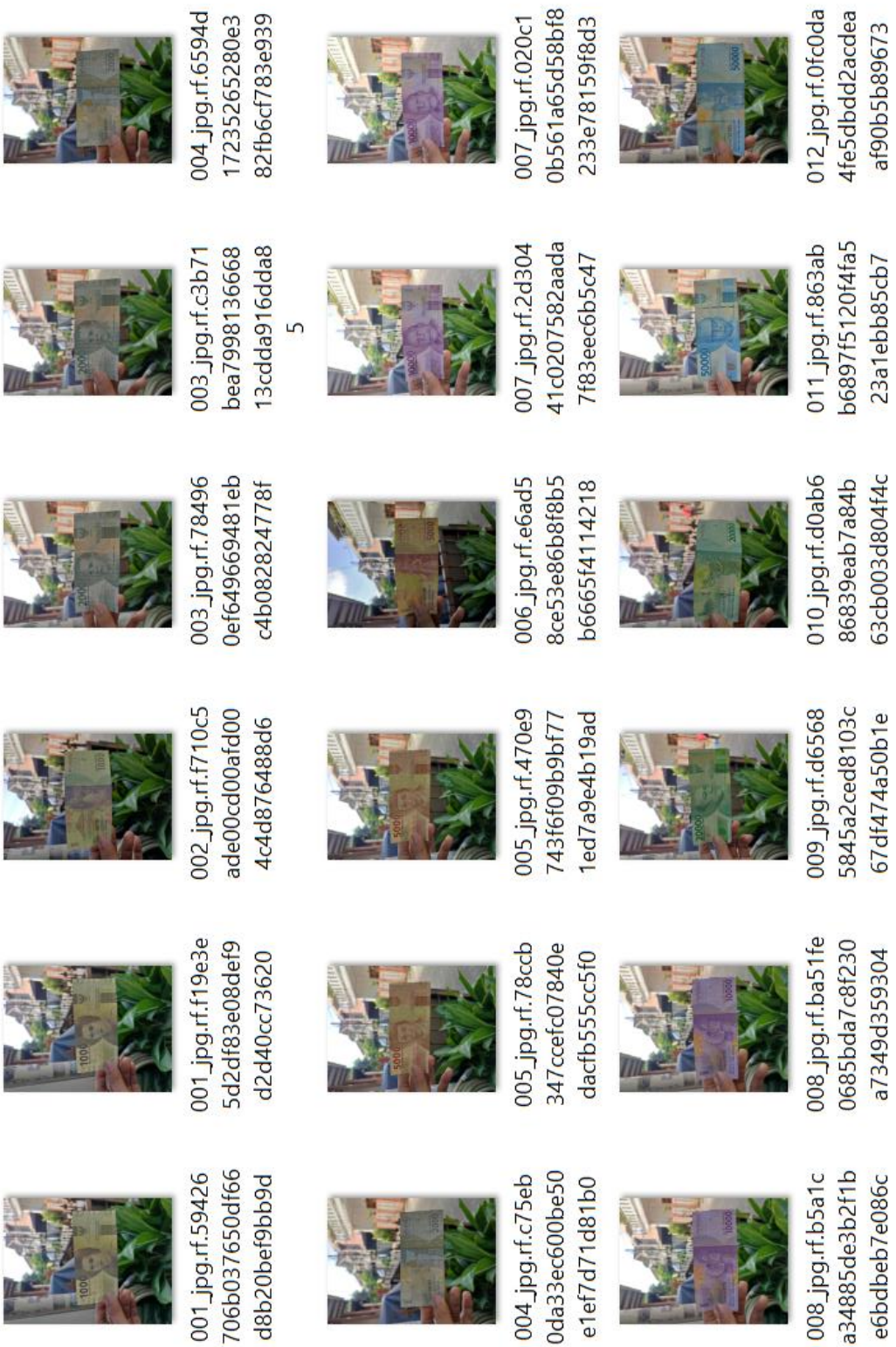
Yuwono, Imam & Mirnawati. 2021. Aksesibilitas Bagi Penyandang Tunanetra Di Lingkungan Lahan Basah. Yogyakarta: Deepublish.

Zhao, Z.-Q., Zheng, P., Xu, S., Wu, X., 2019. Object Detection with Deep Learning: A Review.

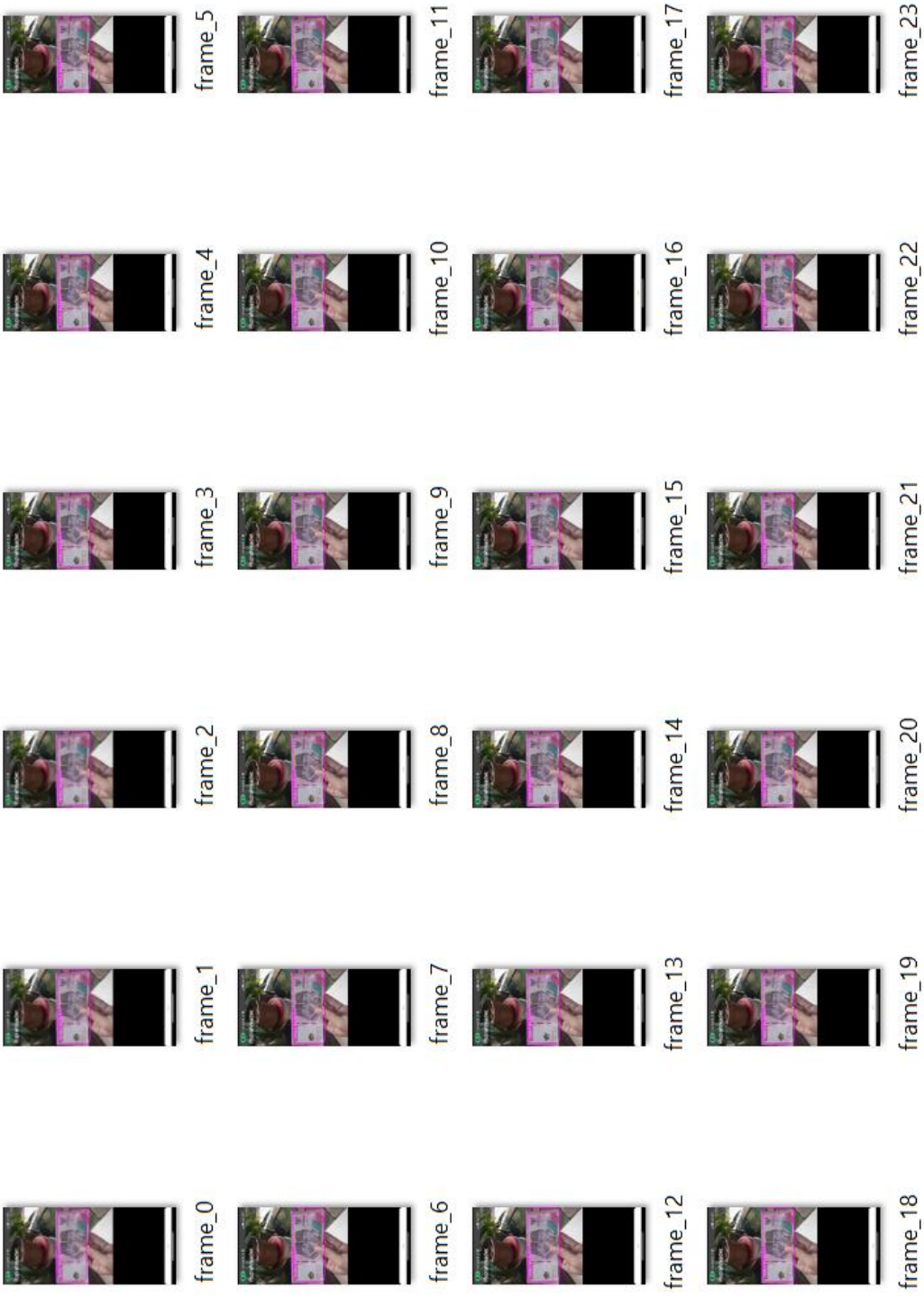
Lampiran 1 Contoh data primer



Lampiran 2 Contoh data sekunder



Lampiran 3 Contoh hasil pengujian



Lampiran 4 Program pelatihan model YOLOv5s

```
# clone YOLOv5 repository
!git clone https://github.com/ultralytics/yolov5  # clone repo
%cd yolov5
# install dependencies
!pip install -qr requirements.txt  # install dependencies (ignore errors)
import torch


from IPython.display import Image, clear_output  # to display images
from utils.downloads import attempt_download  # to download models/datasets


# clear_output()
print('Setup complete. Using torch %s %s' % (torch.__version__,
torch.cuda.get_device_properties(0) if torch.cuda.is_available() else 'CPU'))


#get data from Roboflow
!pip install roboflow


from roboflow import Roboflow
rf = Roboflow(api_key="dHQcLBFTcfdspz3DMQPw")
project = rf.workspace("agil-skripsi-3").project("fix-indonesia-rupiah-banknote-7-
cls")
dataset = project.version(2).download("yolov5")


#custom yaml
%%writetemplate /content/yolov5/models/custom_yolov5s.yaml


# parameters
nc: 7  # number of classes
depth_multiple: 0.33  # model depth multiple
width_multiple: 0.50  # layer channel multiple
```

```
# anchors
anchors:
  - [10,13, 16,30, 33,23]  # P3/8
  - [30,61, 62,45, 59,119]  # P4/16
  - [116,90, 156,198, 373,326]  # P5/32

# YOLOv5 backbone
backbone:
  # [from, number, module, args]
  [[-1, 1, Focus, [64, 3]],  # 0-P1/2
   [-1, 1, Conv, [128, 3, 2]],  # 1-P2/4
   [-1, 3, BottleneckCSP, [128]],
   [-1, 1, Conv, [256, 3, 2]],  # 3-P3/8
   [-1, 9, BottleneckCSP, [256]],
   [-1, 1, Conv, [512, 3, 2]],  # 5-P4/16
   [-1, 9, BottleneckCSP, [512]],
   [-1, 1, Conv, [1024, 3, 2]],  # 7-P5/32
   [-1, 1, SPP, [1024, [5, 9, 13]]],
   [-1, 3, BottleneckCSP, [1024, False]],  # 9
  ]

# YOLOv5 head
head:
  [[-1, 1, Conv, [512, 1, 1]],
   [-1, 1, nn.Upsample, [None, 2, 'nearest']],
   [[-1, 6], 1, Concat, [1]],  # cat backbone P4
   [-1, 3, BottleneckCSP, [512, False]],  # 13

   [-1, 1, Conv, [256, 1, 1]],
   [-1, 1, nn.Upsample, [None, 2, 'nearest']],
   [[-1, 4], 1, Concat, [1]],  # cat backbone P3
   [-1, 3, BottleneckCSP, [256, False]],  # 17 (P3/8-small)
```

```
   [-1, 1, Conv, [256, 3, 2]],
   [[-1, 14], 1, Concat, [1]],  # cat head P4
   [-1, 3, BottleneckCSP, [512, False]],  # 20 (P4/16-medium)

   [-1, 1, Conv, [512, 3, 2]],
   [[-1, 10], 1, Concat, [1]],  # cat head P5
   [-1, 3, BottleneckCSP, [1024, False]],  # 23 (P5/32-large)

   [[17, 20, 23], 1, Detect, [nc, anchors]],  # Detect(P3, P4, P5)
  ]


#train model YOLOv5s
%%time
%cd /content/yolov5/
!python train.py --img 640 --batch 16 --epochs 50 --data
{dataset.location}/data.yaml --cfg ./models/custom_yolov5s.yaml --weights " --
name 1_data  --cache


#show evaluation
from utils.plots import plot_results  # plot results.txt as results.png
Image(filename='/content/yolov5/runs/train/1_data/results.png', width=1000)  #
view results.png
```

Lampiran 5 Program aplikasi sistem deteksi

Program aplikasi sistem deteksi dapat dicek pada laman :
https://github.com/agiltandiera/rupiahreader.git

Lampiran 6 Program untuk ekstrak video menjadi frame

```
import cv2
import os

video_filename = "seratus.mp4"

# save frame
output_folder = "seratus"

# frame limit
desired_frame_count = 25

# make dir if necessary
if not os.path.exists(output_folder):
    os.makedirs(output_folder)

# read video
cap = cv2.VideoCapture(video_filename)

# extract frame and save
count = 0
while cap.isOpened() and count < desired_frame_count:
    ret, frame = cap.read()
    if not ret:
        break
    cv2.imwrite(os.path.join(output_folder, f"frame_{count}.jpg"), frame)
    count += 1

cap.release()
cv2.destroyAllWindows()
```