

## 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. Cikutusi 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



161B20H2.jpg.rf.  
c03ca604372477  
a4df3ed43310a9  
8799



161B20H1.jpg.rf.  
0a2468e8e3745f  
d957b191893b8  
25f7c



161B15H5.jpg.rf.  
7da4d9ad25ee4  
36a5561eaf0de2  
6a257



161B15H3.jpg.rf.  
6c2331008ab584  
50654b6784d92  
ea0c2



161B10H4.jpg.rf.  
6bb4c6fc5e76c2  
83eff9fa761e832  
675



161B10H2.jpg.rf.  
984c2fafed35d1  
0dfe9cdd026a81  
9640



161D15H2.jpg.rf  
.b3c281771d434  
060878da91019  
22db46



161D10H5.jpg.rf  
.ac778ff5038223  
dcd159ad4c1943  
09e7



161D10H4.jpg.rf  
.4aaceca540c2f3  
a90611510456b  
51385



161D10H2.jpg.rf  
.4af81775e11f10  
e490b96a9ae937  
ee19



161B20H5.jpg.rf.  
a0beee31b152c3  
34527f7df6d829  
5ec4



161B20H3.jpg.rf.  
ee7b43b6782f06  
a025af8bb1c0a8  
d344



161D20H3.jpg.rf  
.3d75b8b1737ee  
d85705d3be513  
bf6a92



161D20H2.jpg.rf  
.56461e6279d00  
7c99a69477916e  
58d48



161D20H1.jpg.rf  
.1de33cebfff4f06  
929ed6707611d  
5628



161D15H5.jpg.rf  
.6a9b4a347da19  
143fd1b95bb0f3  
47e67



161D15H4.jpg.rf  
.d07a743b445f4  
a280db4e51542  
715ebe



161D15H3.jpg.rf  
.3a1bb34fcca283  
daedc73f3cb770  
c032

## Lampiran 2 Contoh data sekunder



001\_jpg.rf.59426  
706b037650df66  
d8b20bef9bb9d



001\_jpg.rf.f19e3e  
5d2df83e08def9  
d2d40cc73620



002\_jpg.rf.f710c5  
ade00cd00afd00  
4c4d876488d6



003\_jpg.rf.78496  
0ef649669481eb  
c4b082824778f



003\_jpg.rf.c3b71  
bea7998136668  
13cdda916dda8



004\_jpg.rf.6594d  
17235265280e3  
82fb6cf783e939



004\_jpg.rf.c75eb  
0da33ec600be50  
e1ef7d71d81b0



005\_jpg.rf.78ccb  
347ccef07840e  
dacfb555cc5f0



005\_jpg.rf.470e9  
743f6f09b9bf77  
1ed7a9e4b19ad



006\_jpg.rf.e6ad5  
8ce53e86b8f8b5  
b665f4114218



007\_jpg.rf.2d304  
41c0207582aada  
7f83eec6b5c47



007\_jpg.rf.020c1  
0b561a65d58bf8  
233e78159f8d3



008\_jpg.rf.b5a1c  
a34885de3b2f1b  
e6bdb7e086c



008\_jpg.rf.ba51fe  
0685bda7c8f230  
a7349d359304



009\_jpg.rf.d6568  
5845a2ced8103c  
67df474a50b1e



010\_jpg.rf.d0ab6  
86839eab7a84b  
63cb003d804f4c

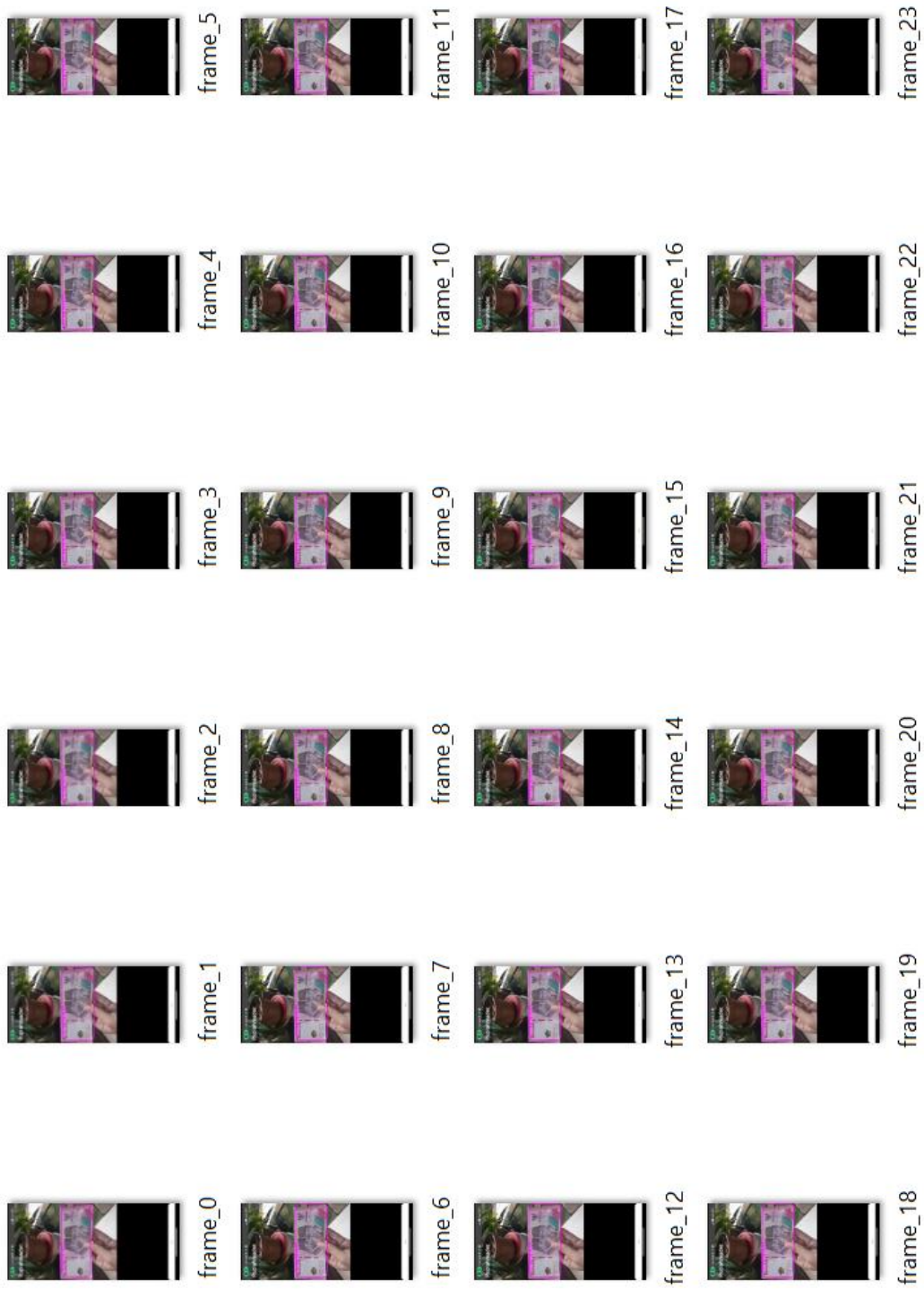


011\_jpg.rf.863ab  
b6897f5120f4fa5  
23a1ebb85cb7



012\_jpg.rf.0fc0da  
4fe5dbdd2acdea  
af90b5b89673

## 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()
```