

DAFTAR PUSTAKA

- Alorang, I. G., Mokolensang, J. F., Watung, J. C., Sinjal, H. J., Monijung, R. D., & Mudeng, J. D. (2023). Substitusi tepung ikan dengan Maggot (*Hermetia illucens*) terhadap pertumbuhan dan efisiensi ikan Nila (*Oreochromis niloticus*). *e-Journal BUDIDAYA PERAIRAN*, 11(2), 198-212.
- Aini, Q., Lutfiani, N., Kusumah, H., & Zahran, M. S. (2021). Deteksi dan Pengenalan Objek Dengan Model Machine Learning: Model Yolo. *CESS (Journal of Computer Engineering, System and Science)*, 6(2), 192.
- Arrahma, S. A., & Mukhaiyar, R. (2023). Pengujian Esp32-Cam Berbasis Mikrokontroler ESP32. *JTEIN: Jurnal Teknik Elektro Indonesia*, 4(1), 60-66.
- Berampu, L. E., Patriono, E., & Amalia, R. (2021). Pemberian kombinasi maggot dan pakan komersial untuk efektifitas pemberian pakan tambahan benih ikan Lele Sangkuriang (*Clarias gariepinus*) oleh kelompok pembudidaya ikan Lele. *Sriwijaya Bioscientia*, 2(2), 35-44.
- Miftah, Z., & Sari, I. P. (2020). Analisis Sistem Pembelajaran Daring Menggunakan Metode Sus. *Research and Development Journal of Education*, 1(1), 40-48.
- Kuria, K. P., Robinson, O. O., & Gabriel, M. M. (2020). Monitoring temperature and humidity using Arduino Nano and Module-DHT11 sensor with real time DS3231 data logger and LCD display.
- Tindage, J., Mokolensang, J. F., Monijung, R. D., Lumenta, C., Mudeng, J. D., & Ngangi, E. L. (2023). Substitusi Tepung Ikan Dengan Maggot (*Hermetia illucens*) Terhadap Efisiensi Dan Pertumbuhan Ikan Mas (*Cyprinus carpio* L). *e-Journal BUDIDAYA PERAIRAN*, 11(2), 119-129.
- Pratama, N. A., Sutrisno, I., & Rinanto, N. (2024). Penerapan Logika Fuzzy untuk Meningkatkan Efisiensi Pengembangbiakan Maggot Berbasis IoT. *Jurnal Elektronika dan Otomasi Industri*, 11(1), 218-226.

- Pambudi, H. K., Kusuma, P. G. A., Yulianti, F., & Julian, K. A. (2020). Prediksi Status Pengiriman Barang Menggunakan Metode Machine Learning. *Jurnal Ilmiah Teknologi Infomasi Terapan*, 6(2), 100-109.
- Rahma, L., Syaputra, H., Mirza, A. H., & Purnamasari, S. D. (2021). Objek Deteksi Makanan Khas Palembang Menggunakan Algoritma YOLO (You Only Look Once). *Jurnal Nasional Ilmu Komputer*, 2(3), 213-232.
- Saragih, G. M., Marhadi, M., Herawati, P., Suzana, A., & Sari, L. C. (2023). Analisis Pemanfaatan Sampah Organik Sebagai Media Perkembangbiakan Maggot. *Jurnal Daur Lingkungan*, 6(1), 22-28.
- Suciati, R. (2017). Efektifitas media pertumbuhan maggots *Hermetia illucens* (lalat tentara hitam) sebagai solusi pemanfaatan sampah organik. *Biosfer: Jurnal Biologi Dan Pendidikan Biologi*, 2(1), 8-13.
- Septiawati, R., Astriani, D., & Ariffianto, M. A. (2021). Pemberdayaan ekonomi masyarakat melalui pengembangan potensi lokal budidaya Black Soldier Fly (maggot) di Desa Sukaratu Karawang. *Al-Kharaj: Jurnal Ekonomi, Keuangan & Bisnis Syariah*, 3(2), 219-229.
- Sulistia, S., & Ambarsari, H. (2021). Deodorisasi Sludge Limbah Industri Makanan untuk Pakan Maggot BSF (Black Soldier Fly) dengan Teknik Biosorpsi. *Jurnal Teknologi Lingkungan*, 22(2), 222-230.

LAMPIRAN

Lampiran 1. Program Arduino Telegram



The screenshot shows the Arduino IDE interface with a project named 'projek_maggot_kondisi' on an 'Arduino Uno' board. The code is written in C++ and is designed to interact with a Telegram bot. It includes comments in Indonesian and handles various commands like '/start', '/flash_led', and '/foto'. The code uses the 'Bot' library for Telegram communication and the 'DigitalWrite' function for controlling an LED. The output window at the bottom shows the status 'Downloaded index package_mpy32_index.json'.

```

120 // Print the received message
121 String text = bot.messages[0].text;
122 Serial.println(text);
123 String welcome = "";
124 String from_name = bot.messages[0].from_name;
125 if (text == "/start") {
126   welcome = "Selamat Datang , " + from_name + "!";
127   welcome += "Gunakan perintah di bawah untuk berinteraksi !\n";
128   welcome += "/foto : Mengambil foto\n";
129   welcome += "/flash_led : Menyalakan LED\n";
130   welcome += "/lampu : Menyalakan lampu\n";
131   welcome += "/Monitoring : Menampilkan data terkini\n";
132   bot.sendMessage(chat_id, welcome, "");
133 }
134 if (text == "/flash_led") {
135   flashState = !flashState;
136   digitalWrite(LED_PIN, flashState);
137   Serial.println("Change flash LED state");
138   welcome += "Berl ESP32-CAM :!\n";
139   welcome += "LED flash :!\n";
140   welcome += "/start : untuk melihat beberapa perintah.\n";
141   bot.sendMessage(chat_id, welcome, "");
142 }
143 if (text == "/foto") {
144   sendPhoto = true;
145   Serial.println("New photo request");
146   welcome += "Berl ESP32-CAM :!\n";
147   welcome += "Mengambil foto :!\n";

```

Lampiran 2. Machine Learning

YOLOv5-Custom-Training.ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text Copy to Drive

Connect on + Gemini

Step 1: Install Requirements

```

# 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'}")

```

Cloning into 'yolov5'...

remote: Enumerating objects: 9403, done.

remote: Counting objects: 100% (27/27), done.

remote: Compressing objects: 100% (12/12), done.

remote: Total 9403 (delta 15), reused 25 (delta 15), pack-reused 9430

Receiving objects: 100% (9403/9403), 9.88 MiB | 22.44 MiB/s, done.

Resolving deltas: 100% (6574/6574), done.

/content/yolov5

636 KB 15.3 MB/s

Lampiran 3. Proses Training data

YOLOv5-Custom-Training.ipynb

File Edit View Insert Runtime Tools Help

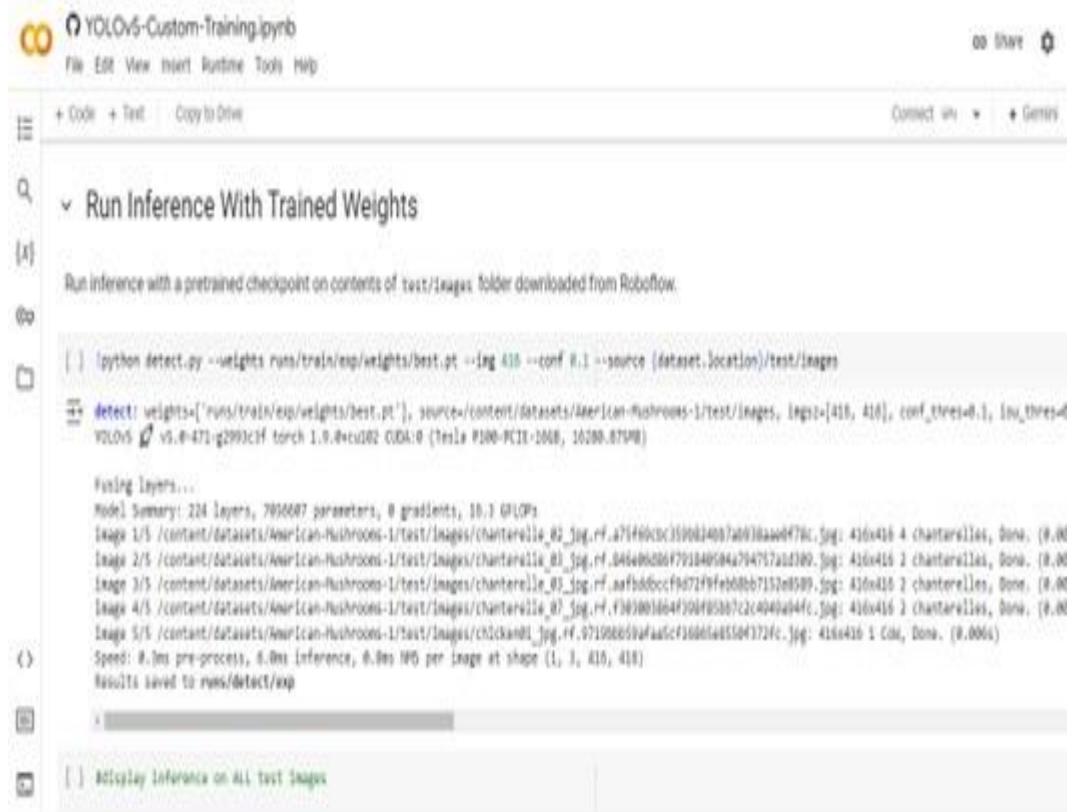
Copy to Drive

Connect to a GPU

Overriding model.yaml nc=80 with nc=3

	from	n	params	module	arguments
0	-1	1	3520	models.common.Focus	[3, 32, 1]
1	-1	1	18560	models.common.Conv	[32, 64, 1, 2]
2	-1	1	18016	models.common.C1	[64, 64, 1]
3	-1	1	73888	models.common.Conv	[64, 128, 3, 2]
4	-1	1	166028	models.common.C1	[128, 128, 1]
5	-1	1	295424	models.common.Conv	[128, 256, 3, 2]
6	-1	1	625152	models.common.C1	[256, 256, 1]
7	-1	1	1180672	models.common.Conv	[256, 512, 3, 2]
8	-1	1	860896	models.common.SPP	[512, 512, [5, 9, 13]]
9	-1	1	1182720	models.common.C1	[512, 512, 1, False]
10	-1	1	151584	models.common.Conv	[512, 256, 1, 1]
11	-1	1	0	torch.nn.modules.upconviling.Upconviling	[None, 2, 'nearest']
12	[-1, 6]	1	0	models.common.Concat	[1]
13	-1	1	362984	models.common.C1	[512, 256, 1, False]
14	-1	1	13824	models.common.Conv	[256, 128, 1, 1]
15	-1	1	0	torch.nn.modules.upconviling.Upconviling	[None, 2, 'nearest']
16	[-1, 4]	1	0	models.common.Concat	[1]
17	-1	1	90080	models.common.C1	[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.C1	[256, 256, 1, False]
21	-1	1	390136	models.common.Conv	[256, 256, 3, 2]
22	[-1, 18]	1	0	models.common.Concat	[1]
23	-1	1	1182720	models.common.C1	[512, 512, 1, False]
24	[-1, 20]	1	0	models.common.Concat	[1]

Lampiran 4. Proses Pendeteksian Objek



```

YOLOv5-Custom-Training.ipynb
File Edit View Insert Runtime Tools Help

+ Code + Text Copy to Drive Connect via + Gemini

Run Inference With Trained Weights

Run inference with a pretrained checkpoint on contents of test/images folder downloaded from Roboflow.

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

detect: weights=['runs/train/exp/weights/best.pt'], source=/content/datasets/American-Mushrooms-1/test/images, imgs=[416, 416], conf_thres=0.1, iou_thres=0.45
YOLOv5 v5.0-471-g2993c3f torch 1.9.0+cu102 CUDA:0 (Tesla P100-PCIE-16GB, 16380.875MB)

Fusing layers...
Model Summary: 224 layers, 705607 parameters, 0 gradients, 10.1 GiB
Image 1/5 /content/datasets/American-Mushrooms-1/test/images/chanterelle_02.jpg.rf.a75f60c3596834087a0818a0d476c.jpg: 416x416 4 chanterelles, Done. (0.006s)
Image 2/5 /content/datasets/American-Mushrooms-1/test/images/chanterelle_03.jpg.rf.046e8d86f791840584a794757a2d380.jpg: 416x416 1 chanterelle, Done. (0.006s)
Image 3/5 /content/datasets/American-Mushrooms-1/test/images/chanterelle_03.jpg.rf.aafbd8ccf9d72f9f6b8bb7157e0589.jpg: 416x416 2 chanterelles, Done. (0.006s)
Image 4/5 /content/datasets/American-Mushrooms-1/test/images/chanterelle_07.jpg.rf.f3e78016e4f70d485b67c2c404a04fc.jpg: 416x416 1 chanterelle, Done. (0.006s)
Image 5/5 /content/datasets/American-Mushrooms-1/test/images/chickandl.jpg.rf.071988659afaa5cf36d6a855047372fc.jpg: 416x416 1 Cde, Done. (0.006s)
Speed: 8.0ms pre-process, 8.0ms inference, 8.0ms NMS per image at shape (1, 3, 416, 416)
Results saved to runs/detect/exp

| | %display inference on all test images

```