AI & SUSTAINABILITY IN VET EDUCATION

Όνομα συγγραφέα/ων: GR Team

Τίτλος WP: WP3 – AI-SAVES@ProjectHub

Αναφορά δραστηριότητας: 3.2 – Πρότυπο έργου για το AI-SAVES@ProjectHub

Λογότυπο:

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Πίνακας περιεχομένων

Πίνακας περιεχομένων **2**

1. Έργο Τίτλος 3

2. Έργο Κατηγορία 3

3. Έργο Περιγραφή 3

4. Λέξεις-κλειδιά 4

5. Ομάδα- στόχος 4

6. Έργο Στόχοι 4

7. Έργο Χρονοδιάγραμμα 4

8. Πόροι παρεχόμενοι 5

9. Εργαλεία και τεχνολογίες που χρησιμοποιούνται 5

10. Βήμα προς βήμα υλοποίησης σχέδιο 5

11. Αξιολόγηση κριτήρια 9

12. Γλώσσες διαθέσιμες 9

13. Συντελεστές 9

14. Ανατροφοδότηση και συνεχής βελτίωση 10

Υποβολής σημειώσεις **10**

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

AI-SAVES@PROJECTHUB

---

1. Τίτλος έργου:

Smart Bird AI – Έξυπνο Σύστημα Ανίχνευσης Πτηνών

---

2. Κατηγορία έργου:

AI / Computer Vision / Smart Environment

3. Περιγραφή έργου:

Το έργο παρουσιάζει ένα έξυπνο σύστημα παρακολούθησης πτηνών με υποστήριξη AI, το οποίο ανιχνεύει και αναγνωρίζει αυτόματα πτηνά χρησιμοποιώντας κάμερα και Raspberry Pi.
Με τη χρήση computer vision και ενός ελαφρού AI model, το σύστημα καταγράφει εικόνες πτηνών που επισκέπτονται μια ταΐστρα και τα ταξινομεί.
Το έργο δείχνει πώς η τεχνητή νοημοσύνη μπορεί να αξιοποιηθεί για την παρακολούθηση της άγριας ζωής, την εκπαίδευση στη βιοποικιλότητα και την περιβαλλοντική STEM μάθηση.

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

4. Λέξεις-κλειδιά:

AI, Computer Vision, Ανίχνευση Πτηνών, Raspberry Pi, Παρακολούθηση Άγριας Ζωής, Smart Environment

---

5. Ομάδα-στόχος:

Μαθητές/τριες VET, μαθητές/τριες δευτεροβάθμιας εκπαίδευσης, STEM clubs, αρχάριοι AI developers

---

6. Στόχοι έργου:

- Κατανόηση των βασικών αρχών του computer vision και της ταξινόμησης εικόνων με AI.
- Εκμάθηση της σύνδεσης ενός camera module σε microcomputer.
- Υλοποίηση ενός AI model για την ανίχνευση πτηνών.
- Αποθήκευση και ανάλυση δεδομένων παρατήρησης.
- Διερεύνηση εφαρμογών του AI στην περιβαλλοντική παρακολούθηση.

---

7. Χρονοδιάγραμμα έργου:

Ημέρα 1 – Εισαγωγή στο AI και στην παρακολούθηση άγριας ζωής
Ημέρα 2 – Ρύθμιση hardware (Raspberry Pi και camera)
Ημέρα 3 – Εγκατάσταση software και AI libraries
Ημέρα 4 – Εκτέλεση του bird detection model
Ημέρα 5 – Καταγραφή και οπτικοποίηση δεδομένων

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

8. Παρεχόμενοι πόροι:

- Raspberry Pi
- Raspberry Pi Camera Module
- Ταΐστρα πτηνών ή εξωτερικό σημείο παρακολούθησης
- Κάρτα microSD με Raspberry Pi OS
- Σύνδεση στο διαδίκτυο

---

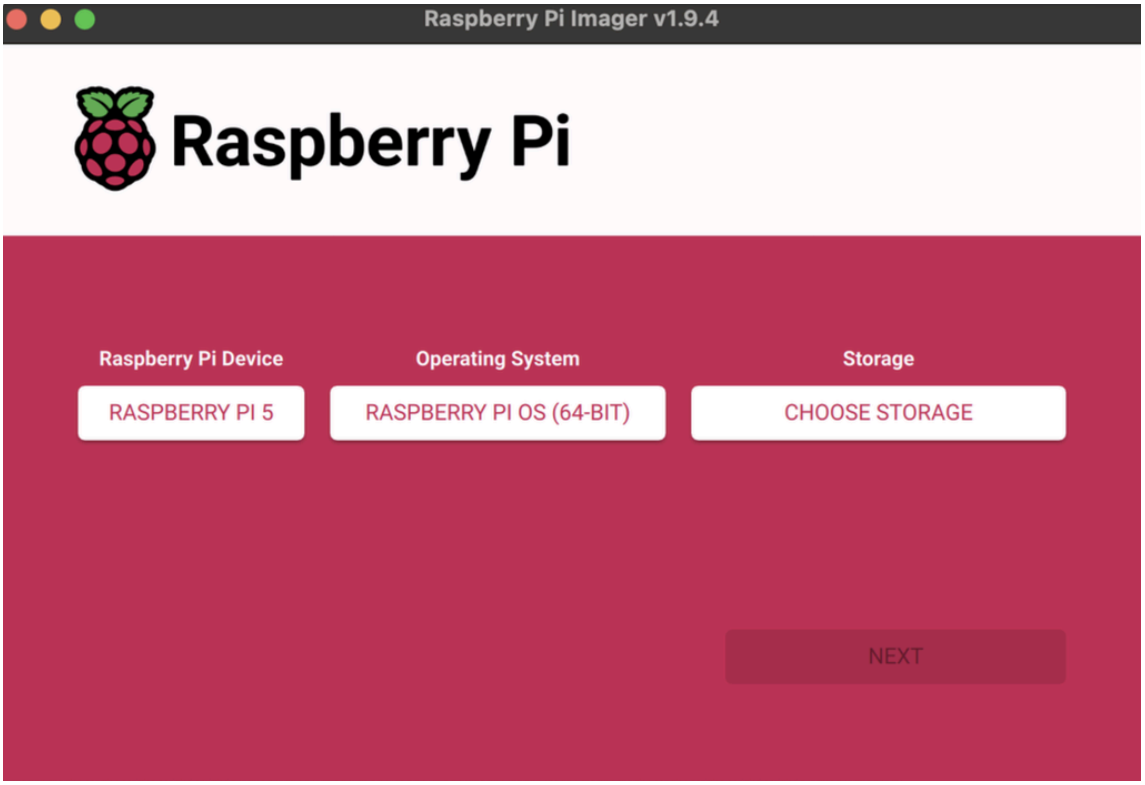
9. Εργαλεία και τεχνολογίες που χρησιμοποιούνται:

- Raspberry Pi
- Python
- OpenCV
- TensorFlow Lite
- Camera Module
- Προαιρετικό cloud dashboard για τις παρατηρήσεις

---

10. Βήμα προς βήμα σχέδιο υλοποίησης:

Βήμα 1: Flashing της κάρτας microSD με Raspberry Pi OS



1. Για να εγκαταστήσετε το Raspberry Pi OS, τοποθετήστε την κάρτα microSD

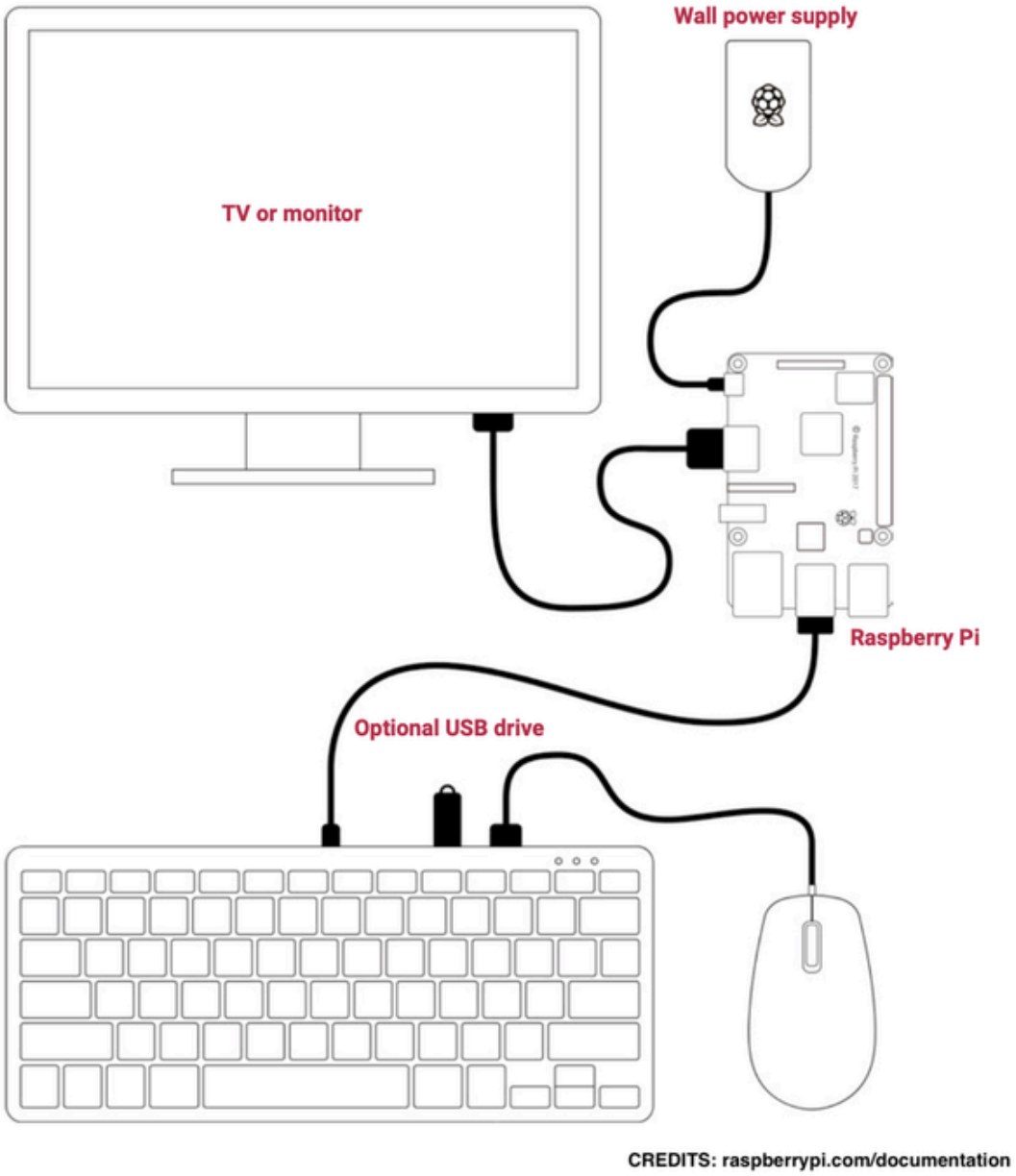
Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

σε έναν USB adapter και συνδέστε τον στον υπολογιστή σας.

2. Κατεβάστε και εγκαταστήστε το software Raspberry Pi Imager από την επίσημη ιστοσελίδα του Raspberry Pi.
3. Ανοίξτε το Imager και ρυθμίστε το σύμφωνα με τις επιλογές που φαίνονται στην εικόνα.
4. Επιλέξτε Write για να γίνει flash το OS. Η διαδικασία θα διαρκέσει περίπου 30 λεπτά.
5. Μόλις ολοκληρωθεί η διαδικασία, αφαιρέστε την κάρτα microSD, τοποθετήστε τη στη θύρα microSD του Raspberry Pi και ενεργοποιήστε το. Το Raspberry Pi θα εκκινήσει στο OS.

Βήμα 2: Ρύθμιση του Raspberry Pi



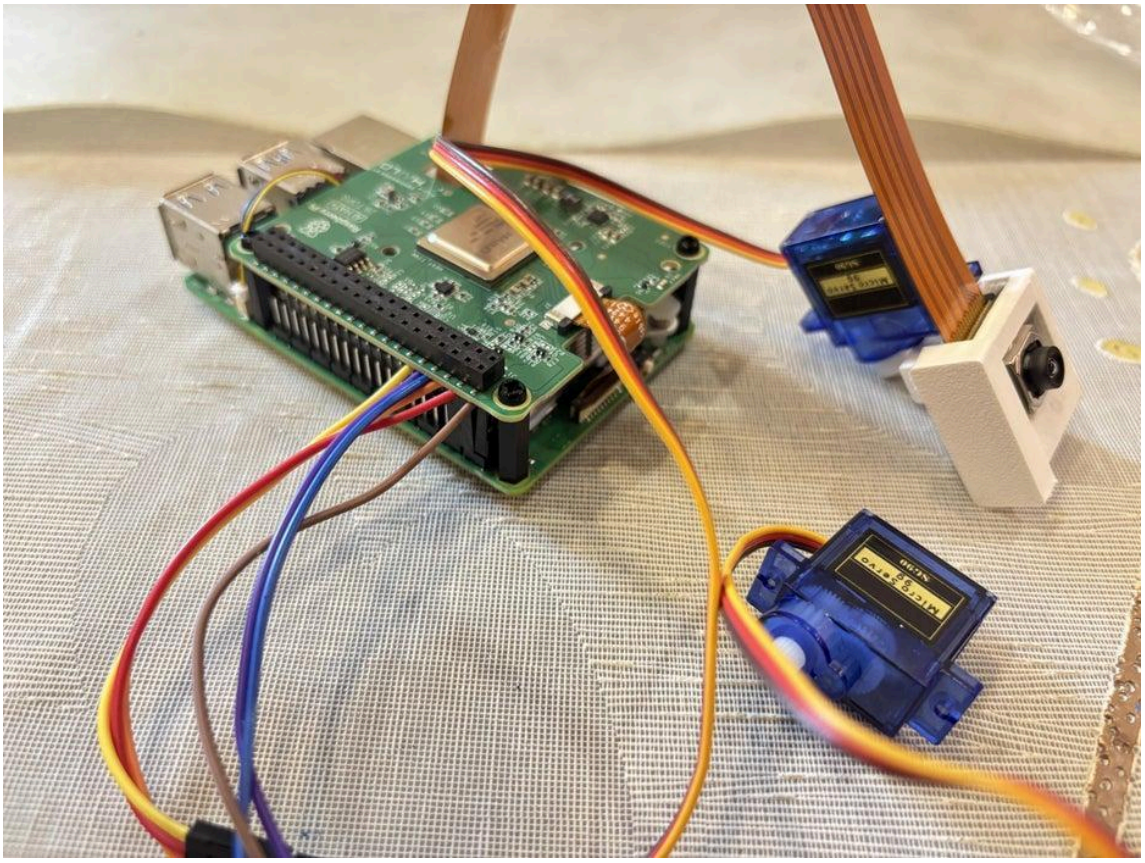
CREDITS: raspberrypi.com/documentation

Το Raspberry Pi είναι ουσιαστικά ένας υπολογιστής με χαμηλότερες προδιαγραφές, αλλά λειτουργεί με τον ίδιο τρόπο όπως ένας υπολογιστής. Απαιτεί ακόμη monitor/οθόνη και συσκευές εισόδου (keyboard & mouse).

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Για να το ρυθμίσετε σωστά, συνδέστε το keyboard και το mouse στο Raspberry Pi μέσω USB 3.0 (μπλε θύρες). Στη συνέχεια, συνδέστε το καλώδιο Micro-HDMI στο Raspberry Pi και την πλευρά HDMI στο monitor. Ενεργοποιήστε το Raspberry Pi, περιμένετε μερικά δευτερόλεπτα και θα εμφανιστεί η σελίδα ρυθμίσεων.
Σε αυτή τη σελίδα θα σας ζητηθεί να εισαγάγετε στοιχεία διαχείρισης, όπως το username και το password του Raspberry Pi, το Wi-Fi SSID και το password, καθώς και προτιμήσεις τοποθεσίας και γλώσσας. Συμπληρώστε αυτές τις πληροφορίες.
Βήμα 3: Σύνδεση της camera και τοποθέτηση του AI Hat Module στο Raspberry Pi 5



Για να ρυθμίσετε το Raspberry Pi 5 με την camera και το AI Hat, ξεκινήστε συνδέοντας την camera. Ανασηκώστε απαλά το μαύρο πλαστικό τμήμα στη θύρα της camera και στη συνέχεια εισαγάγετε το μικρό άκρο του καλωδίου της camera, με τις γυαλιστερές επαφές στραμμένες προς τη σωστή κατεύθυνση. Πιέστε ξανά το μαύρο τμήμα προς τα κάτω για να ασφαλίσει.
Είναι προτιμότερο να συνδέσετε πρώτα την camera, επειδή έτσι έχετε περισσότερο χώρο και η διαδικασία γίνεται ευκολότερη. Στη συνέχεια, τοποθετήστε το AI Hat πάνω στο Raspberry Pi, διασφαλίζοντας ότι οι connectors ευθυγραμμίζονται.
Χρησιμοποιήστε τις κοντές βίδες στο κάτω μέρος του Raspberry Pi και τις μακριές βίδες στο επάνω μέρος για να στερεώσετε το AI Hat. Μόλις το AI Hat τοποθετηθεί, γίνεται πιο δύσκολο να αφαιρέσετε ή να ρυθμίσετε την camera χωρίς πρώτα να αφαιρέσετε το Hat.
Βεβαιωθείτε ότι έχετε συνδέσει και το servo motor στο Raspberry Pi μέσω των GPIO pins. Ακολουθήστε την παρακάτω συνδεσμολογία:

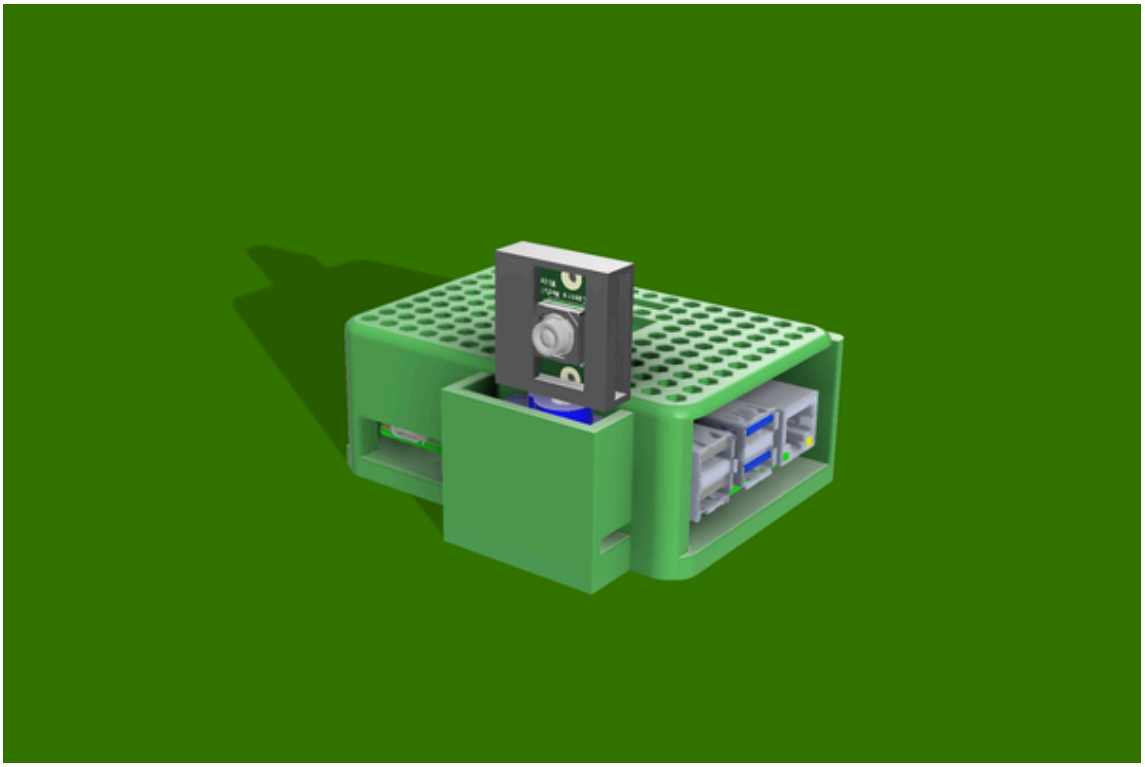
1. Καφέ καλώδιο → GND
2. Κόκκινο καλώδιο → 5V

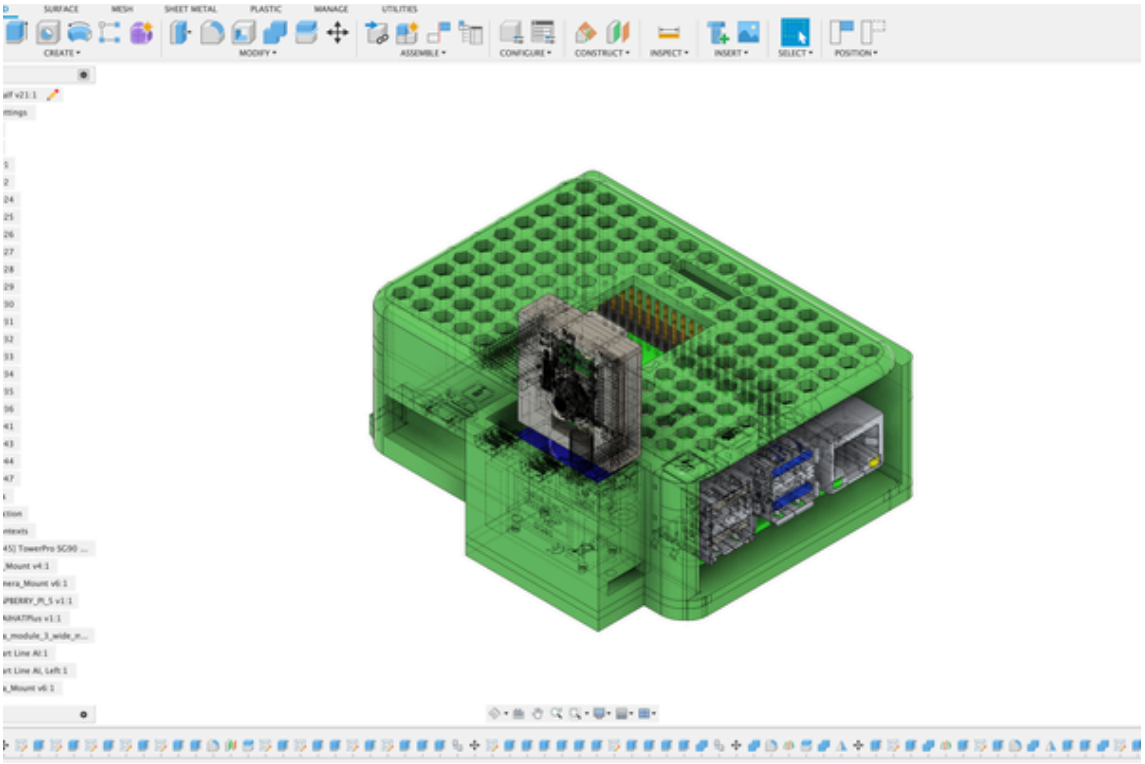
Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

3. Κίτρινο καλώδιο → GPIO 17

Βήμα 4: 3D render του περιβλήματος





Δημιούργησα αυτό το CAD χρησιμοποιώντας Autodesk Fusion 360. Η πρώτη εικόνα είναι ένα render, ενώ η δεύτερη μπορεί να σας βοηθήσει να κατανοήσετε τα συγκεκριμένα εργαλεία που χρησιμοποίησα. Χρησιμοποίησα επίσης το Autodesk για να αξιολογήσω την αντοχή αυτού του μοντέλου και να διασφαλίσω ότι η κατασκευή θα είναι επαρκής.

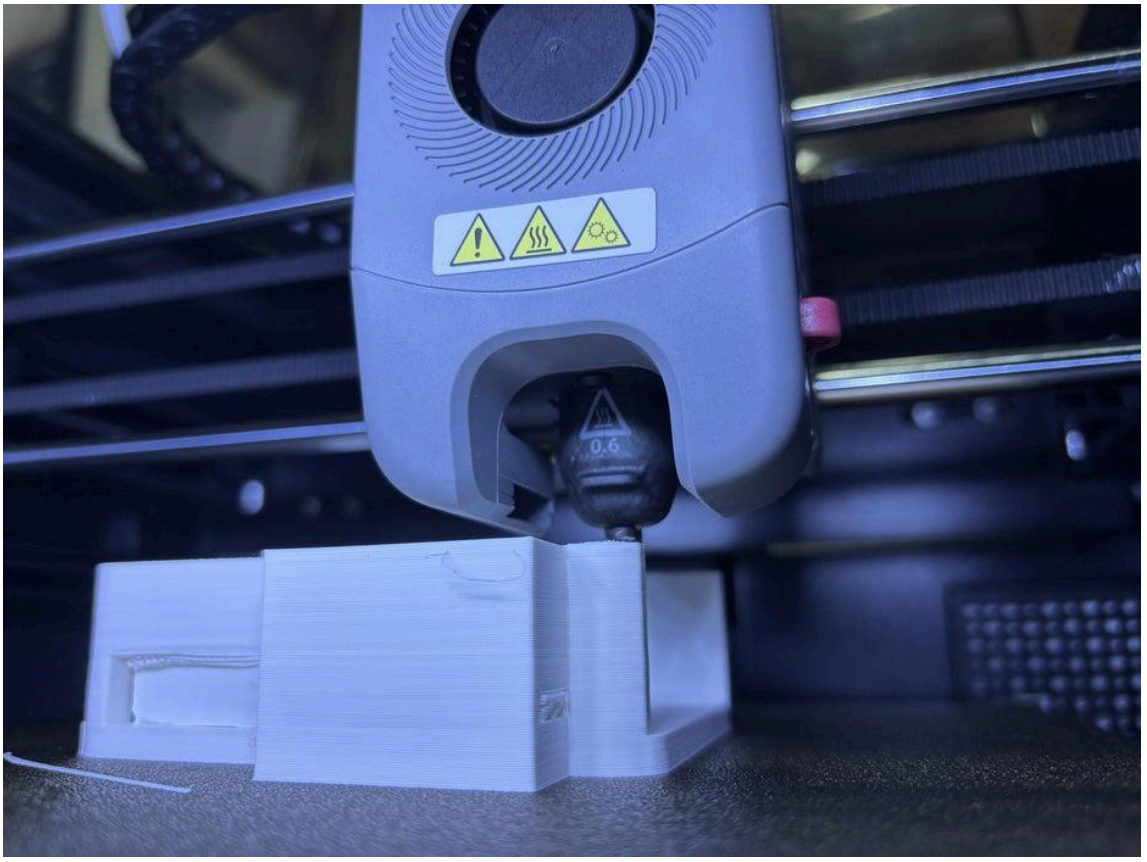
Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Για να δείτε το 3D render του περιβλήματος, ανατρέξτε στο παραπάνω σχέδιο. Αυτό μπορεί επίσης να βοηθήσει στην οπτικοποίηση πιθανών αλλαγών για το έργο σας.

Smart Line Bird, Version 1 v12.stl

Βήμα 5: Ασφάλεια στην 3D Printing

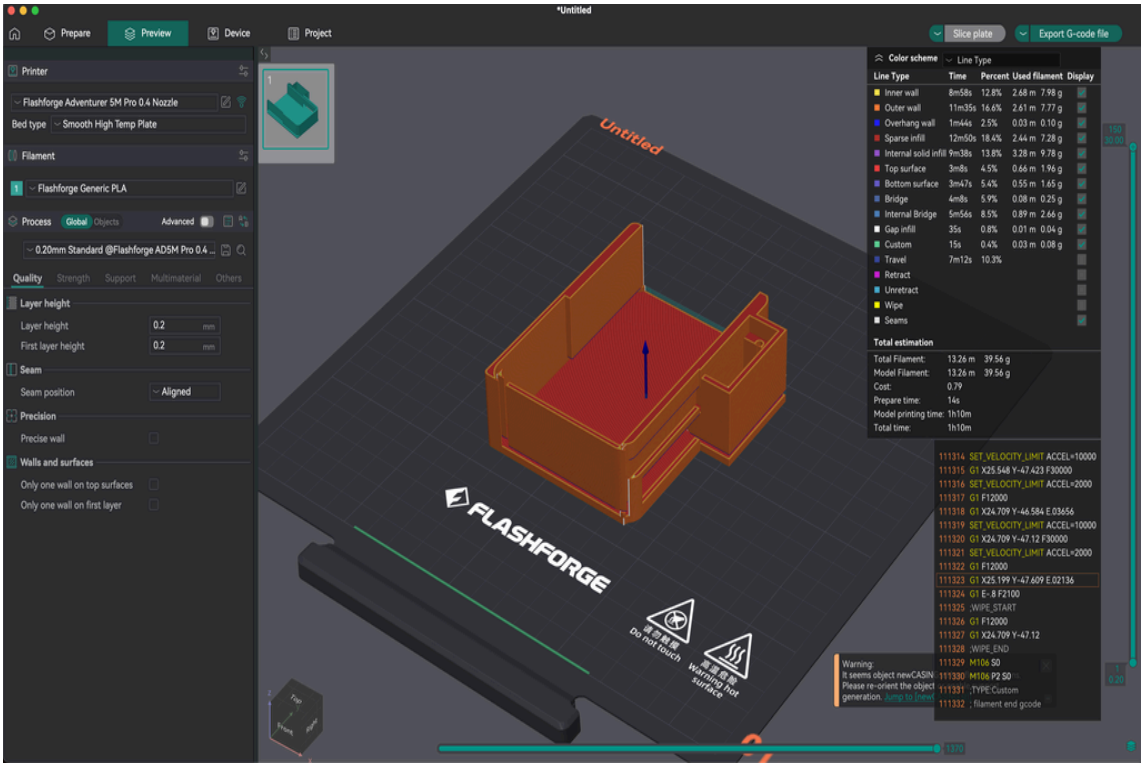


Όταν αφαιρείτε το 3D print από την επιφάνεια εκτύπωσης μετά την ολοκλήρωση της διαδικασίας, ΒΕΒΑΙΩΘΕΙΤΕ ότι περιμένετε 5 λεπτά, ώστε η πλάκα να κρυώσει. Αυτό θα αποτρέψει πιθανούς κινδύνους, όπως εγκαύματα.

Βήμα 6: 3D Printing του περιβλήματος

Co-funded by the European Union

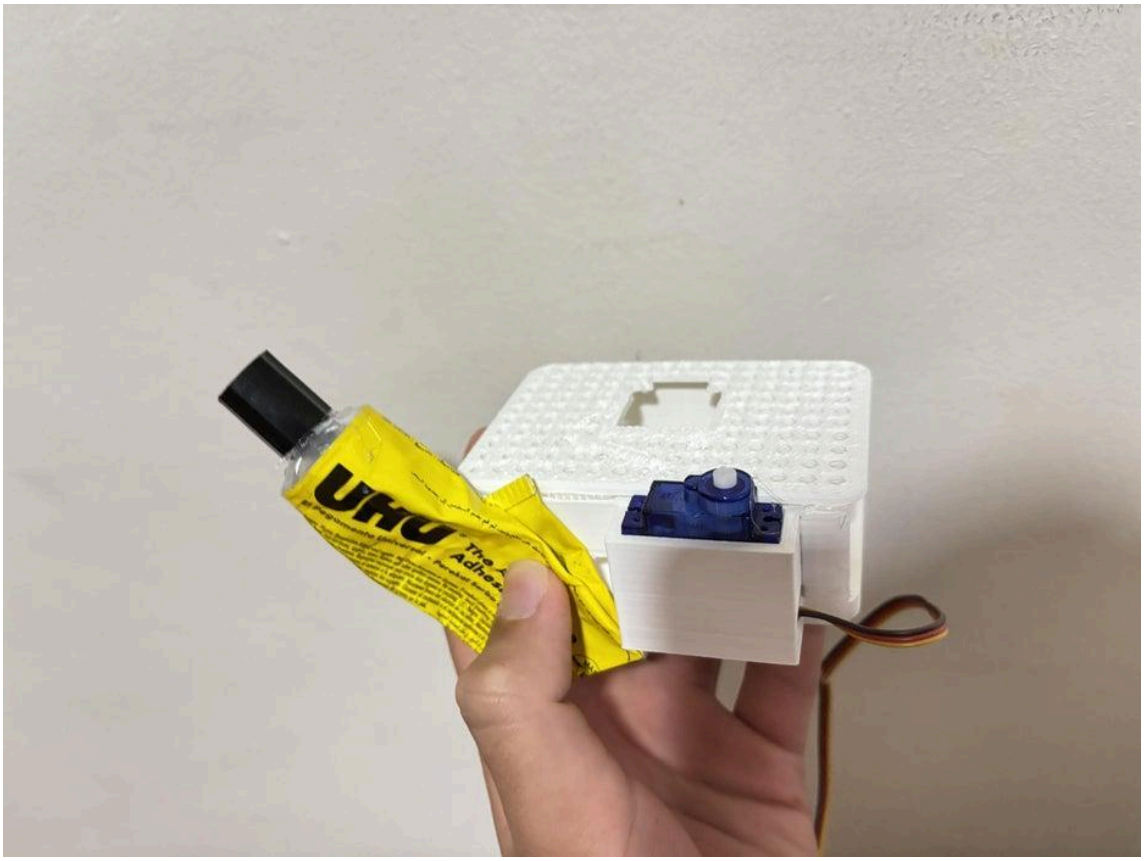
This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



Κατεβάστε όλα τα .STL files που έχω προσθέσει στην ενότητα Supporting Files και εισαγάγετέ τα στο slicer tool σας, όπως το Orca ή το FlashPrint. Για τη συγκεκριμένη διαδικασία, προτείνω το Orca, επειδή μπορείτε να προσαρμόσετε το infill του αρχείου. Αυτό σημαίνει ότι μπορείτε να το ορίσετε στο 100% στις περιοχές γύρω από τις βίδες και χαμηλότερα στα υπόλοιπα σημεία. Δεν προτείνω να το ορίσετε στο 100% παντού, επειδή έτσι το module γίνεται εξαιρετικά βαρύ.
Δείτε την εικόνα όπου έχω ρυθμίσει αυτή την επιλογή στο Orca.
Μετά την ολοκλήρωση της εκτύπωσης, θυμηθείτε να αφαιρέσετε τυχόν supports.
Βήμα 7: Κόλληση

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



Κολλήστε τα μέρη μεταξύ τους χρησιμοποιώντας κόλλα UHU. Ξεκινήστε κολλώντας το επάνω μέρος στο κάτω μέρος του case. Αν έχετε clamps, αυτή είναι κατάλληλη στιγμή να τα χρησιμοποιήσετε· διαφορετικά, λειτουργούν πολύ καλά και τα clips.

Στη συνέχεια, κολλήστε το περίβλημα στη βάση χρησιμοποιώντας την ίδια κόλλα. Εδώ δεν χρησιμοποίησα clamps, επειδή υπήρχε πολύ μεγάλη επιφάνεια για να σταθεροποιηθεί καλά η κόλλα. Η UHU αρχίζει να πήζει μετά από 30 λεπτά, αλλά την άφησα όλο το βράδυ για να βεβαιωθώ ότι θα είναι ανθεκτική.

Βήμα 8: Εγκατάσταση

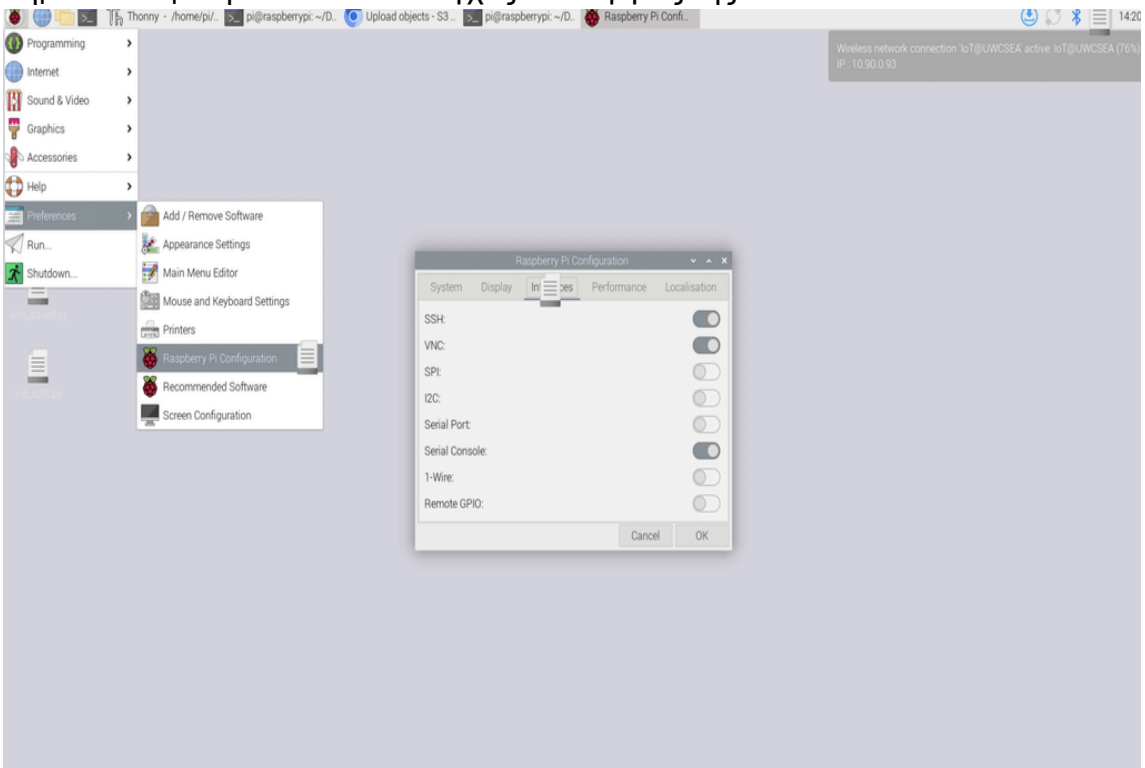
Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



Η εγκατάσταση του module είναι εξαιρετικά απλή: απλώς τοποθετήστε τα components σε εξωτερικό χώρο, κοντά στο πεδίο παρατήρησης.

Βήμα 9: Ρύθμιση του VNC και έλεγχος λειτουργίας της camera



Συνήθως, όταν συνδέεστε στο Raspberry Pi, χρειάζεται να ρυθμίσετε monitor, keyboard και mouse, κάτι που είναι αρκετά άβολο. Υπάρχουν όμως απλούστερες

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

εναλλακτικές, όπως η χρήση ενός εργαλείου που ονομάζεται VNC. Αυτό σας επιτρέπει να αποκτάτε πρόσβαση στο Raspberry Pi χωρίς monitor, keyboard ή mouse - το μόνο που χρειάζεται είναι τροφοδοσία.
Για να το ρυθμίσετε, κάντε κλικ στο λογότυπο του Raspberry Pi, μεταβείτε στο Raspberry Pi Configuration, επιλέξτε Interfaces και ενεργοποιήστε την επιλογή VNC. Στη συνέχεια, μεταβείτε στον υπολογιστή σας και εγκαταστήστε το RealVNC. Αφού ενεργοποιήσετε το VNC, πρέπει να βρείτε την IP address του Raspberry Pi. Μπορείτε να το κάνετε είτε μέσα από το Raspberry Pi UI είτε χρησιμοποιώντας IP scanner. Για να τη βρείτε από το UI, μετακινήστε τον cursor πάνω από το λογότυπο Wi-Fi και θα εμφανιστεί η IP address. Πληκτρολογήστε απλώς αυτούς τους αριθμούς στο RealVNC στον υπολογιστή σας και εισαγάγετε το username και το password. Έπειτα θα έχετε πλήρη έλεγχο μέσω headless setup. Αν θέλετε να χρησιμοποιήσετε IP scanner, αναζητήστε τις διαθέσιμες IPs και εισαγάγετε τον αντίστοιχο αριθμό στο RealVNC.

Βήμα 10: Δοκιμή δεδομένων με code
Τώρα περνάμε στο πιο ενδιαφέρον μέρος - τον code. Αρχικά, πρέπει να συλλέξουμε τα videos που θέλουμε να αναλύσουμε. Προτείνω περίπου 5 ώρες υλικού για αυτό το στάδιο. Μπορείτε να χρησιμοποιήσετε τον παρακάτω code, ο οποίος βοηθά στη συλλογή δεδομένων και στην αποθήκευσή τους σε έναν τοπικό φάκελο με όνομα 'videos'.

Τα videos που θα καταγράψετε σε αυτό το βήμα θα αφορούν απλώς τον εξωτερικό χώρο που σας ενδιαφέρει να παρακολουθήσετε.

```
from picamera2 import Picamera2
from picamera2.encoders import H264Encoder
from picamera2.outputs import FfmpegOutput
from datetime import datetime, timedelta, time as dtime
import os
import time
import sys

def next_window(now):
today = now.date()
start_today = datetime.combine(today, dtime(12, 20, 0))
end_today = datetime.combine(today, dtime(14, 0, 0))

if now < start_today:
return start_today, end_today
elif now < end_today:
return now, end_today
else:
tomorrow = today + timedelta(days=1)
start_tomorrow = datetime.combine(tomorrow, dtime(12, 20, 0))
end_tomorrow = datetime.combine(tomorrow, dtime(14, 0, 0))
return start_tomorrow, end_tomorrow

def main():
out_dir = os.path.join(os.getcwd(), "videos")
os.makedirs(out_dir, exist_ok=True)
```

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

```
now = datetime.now()
start_dt, end_dt = next_window(now)

stamp = start_dt.strftime("%d%m%Y")
filename = f"{stamp}_video.mp4"
out_path = os.path.join(out_dir, filename)

wait_seconds = (start_dt - datetime.now()).total_seconds()
if wait_seconds > 0:
print(f"waiting until {start_dt.strftime('%Y-%m-%d %H:%M:%S')}...")
try:
time.sleep(wait_seconds)
except KeyboardInterrupt:
print("\nInterrupted while waiting. Exiting.")
sys.exit(0)

duration = max(0, (end_dt - datetime.now()).total_seconds())
if duration == 0:
print("[Scheduler] No time left in the window. Exiting.")
return

print(f"Starting capture at {datetime.now().strftime('%H:%M:%S')} "
f"for {int(duration)} seconds (until {end_dt.strftime('%H:%M:%S')}).")
print(f"[Recorder] Output: {out_path}")

picam2 = Picamera2()
video_config = picam2.create_video_configuration(main={"size": (1920, 1080)})
picam2.configure(video_config)

encoder = H264Encoder(bitrate=10_000_000)
output = FfmpegOutput(out_path)

try:
picam2.start_recording(encoder, output)
time.sleep(duration)
except KeyboardInterrupt:
print("\n[Recorder] Interrupted. Stopping recording...")
finally:
try:
picam2.stop_recording()
except Exception:
pass
try:
picam2.close()
except Exception:
pass

print(f"Finished at {datetime.now().strftime('%H:%M:%S')}. Saved: {out_path}")

if __name__ == "__main__":
  main()
```

Αν θέλετε να χρησιμοποιήσετε code που στέλνει ειδοποίηση στο email σας όταν ανιχνεύεται πτηνό, χρησιμοποιήστε τον παρακάτω code. Σημειώστε ότι πρέπει να

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

αντικαταστήσετε τις variables με την κατάλληλη email address και τα αντίστοιχα passwords.

```
import argparse
import os
import smtplib
import ssl
import time
from datetime import datetime
from email.message import EmailMessage
from email.utils import formatdate

import cv2
import numpy as np
import pandas as pd
import torch

SMTP_SERVER = "smtp.gmail.com"
SMTP_PORT = 587
GMAIL_SENDER = "youremail@gmail.com"
GMAIL_APP_PASSWORD = "xxxx xxxx xxxx xxxx"
GMAIL_RECIPIENT = "recipient@example.com"

CONFIDENCE_THRESHOLD = 0.35
NMS_IOU_THRESHOLD = 0.45
SNAPSHOT_DIR = "snapshots"
PREVIEW_WINDOW_NAME = "Bird Watch"

def send_email_with_image(subject, body, image_path):
    msg = EmailMessage()
    msg["Subject"] = subject
    msg["From"] = GMAIL_SENDER
    msg["To"] = GMAIL_RECIPIENT
    msg["Date"] = formatdate(localtime=True)
    msg.set_content(body)
    if image_path and os.path.exists(image_path):
        with open(image_path, "rb") as f:
            data = f.read()
        ext = os.path.splitext(image_path)[1].lower()
        maintype, subtype = ("image", "jpeg") if ext in [".jpg", ".jpeg"] else ("image", "png")
        msg.add_attachment(data, maintype=maintype, subtype=subtype,
filename=os.path.basename(image_path))
    context = ssl.create_default_context()
    with smtplib.SMTP(SMTP_SERVER, SMTP_PORT) as server:
        server.ehlo()
        server.starttls(context=context)
        server.ehlo()
        server.login(GMAIL_SENDER, GMAIL_APP_PASSWORD.replace(" ", ""))
        server.send_message(msg)

def draw_boxes(frame, detections_df, color=(0, 255, 0)):
    for _, row in detections_df.iterrows():
        if row["name"] != "bird":
            continue
        x1, y1, x2, y2 = int(row["xmin"]), int(row["ymin"]), int(row["xmax"]), int(row["ymax"])
        conf = float(row["confidence"])
```

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

```
        label = f"bird {conf:.2f}"
        cv2.rectangle(frame, (x1, y1), (x2, y2), color, 2)
        cv2.putText(frame, label, (x1, max(0, y1 - 6)), cv2.FONT_HERSHEY_SIMPLEX, 0.6, color, 2,
cv2.LINE_AA)
    return frame

def save_snapshot(frame, folder):
    os.makedirs(folder, exist_ok=True)
    ts = datetime.now().strftime("%Y%m%d_%H%M%S")
    path = os.path.join(folder, f"bird_{ts}.jpg")
    cv2.imwrite(path, frame)
    return path

def run(video_path, cooldown_s):
    model = torch.hub.load("ultralytics/yolov5", "yolov5s", pretrained=True)
    model.conf = CONFIDENCE_THRESHOLD
    model.iou = NMS_IOU_THRESHOLD
    model.classes = None
    model.max_det = 300

    cap = cv2.VideoCapture(video_path)
    if not cap.isOpened():
        raise RuntimeError(f"Could not open video file: {video_path}")

    last_email_time = 0.0

    try:
        while True:
            ok, frame = cap.read()
            if not ok:
                break
            results = model(frame, size=640)
            df = results.pandas().xyxy[0]
            bird_df = df[(df["name"] == "bird") & (df["confidence"] >= CONFIDENCE_THRESHOLD)]
            display_frame = frame.copy()
            if not bird_df.empty:
                display_frame = draw_boxes(display_frame, bird_df)
            cv2.imshow(PREVIEW_WINDOW_NAME, display_frame)
            if cv2.waitKey(1) & 0xFF == ord("q"):
                break
            now = time.time()
            if not bird_df.empty and (now - last_email_time >= cooldown_s):
                snapshot_path = save_snapshot(display_frame, SNAPSHOT_DIR)
                count = len(bird_df)
                top_conf = float(bird_df["confidence"].max())
                subject = f"[Bird Alert] {count} bird(s) detected - {datetime.now().strftime('%Y-%m-%d
%H:%M:%S')}"
                body = f"Detected {count} bird(s) with top confidence {top_conf:.2f}.\nSource:
{video_path}\nSnapshot: {os.path.basename(snapshot_path)}\nCooldown: {cooldown_s} seconds\n"
                try:
                    send_email_with_image(subject, body, snapshot_path)
                    last_email_time = now
                except Exception as e:
                    print(f"Email send failed: {e}")
    finally:
        cap.release()
```


This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

```
    cv2.destroyAllWindows()

def parse_args():
  parser = argparse.ArgumentParser()
  parser.add_argument("--video", type=str, default="input.mp4")
  parser.add_argument("--cooldown", type=int, default=180)
  args = parser.parse_args()
  return args.video, args.cooldown

if __name__ == "__main__":
  vid, cd = parse_args()
  run(vid, cd)
```

Αν θέλετε να χρησιμοποιήσετε code που παρέχει επίσης ένα ενδιαφέρον γράφημα με τον αριθμό των πτηνών που εμφανίζονται σε συγκεκριμένες χρονικές στιγμές, χρησιμοποιήστε τον παρακάτω code:

```
import argparse
import os
import csv
from datetime import datetime

import cv2
import pandas as pd
import torch
import matplotlib.pyplot as plt

CONFIDENCE_THRESHOLD = 0.35
NMS_IOU_THRESHOLD = 0.45
CSV_FILE = "bird_counts.csv"

def run(video_path):
  model = torch.hub.load("ultralytics/yolov5", "yolov5s", pretrained=True)
  model.conf = CONFIDENCE_THRESHOLD
  model.iou = NMS_IOU_THRESHOLD

  cap = cv2.VideoCapture(video_path)
  if not cap.isOpened():
    raise RuntimeError(f"Could not open video file: {video_path}")

  fps = cap.get(cv2.CAP_PROP_FPS)
  frame_count = 0

  with open(CSV_FILE, mode="w", newline="") as f:
    writer = csv.writer(f)
    writer.writerow(["time_seconds", "bird_count"])

    while True:
      ok, frame = cap.read()
      if not ok:
        break

      results = model(frame, size=640)
      df = results.pandas().xyxy[0]
```

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

```
        bird_df = df[(df["name"] == "bird") & (df["confidence"] >= CONFIDENCE_THRESHOLD)]
        bird_count = len(bird_df)

        time_sec = frame_count / fps
        writer.writerow([round(time_sec, 2), bird_count])

        display_frame = frame.copy()
        for _, row in bird_df.iterrows():
            x1, y1, x2, y2 = int(row["xmin"]), int(row["ymin"]), int(row["xmax"]), int(row["ymax"])
            conf = float(row["confidence"])
            label = f"bird {conf:.2f}"
            cv2.rectangle(display_frame, (x1, y1), (x2, y2), (0, 255, 0), 2)
            cv2.putText(display_frame, label, (x1, max(0, y1 - 6)),
                        cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 255, 0), 2, cv2.LINE_AA)

        cv2.imshow("Bird Tracking", display_frame)
        if cv2.waitKey(1) & 0xFF == ord("q"):
            break

        frame_count += 1

    cap.release()
    cv2.destroyAllWindows()

    data = pd.read_csv(CSV_FILE)
    plt.figure(figsize=(10, 5))
    plt.plot(data["time_seconds"], data["bird_count"], marker="o")
    plt.xlabel("Time (seconds)")
    plt.ylabel("Number of birds")
    plt.title("Birds detected over time")
    plt.grid(True)
    plt.savefig("bird_graph.png")
    plt.show()

def parse_args():
    parser = argparse.ArgumentParser()
    parser.add_argument("--video", type=str, default="input.mp4")
    return parser.parse_args().video

if __name__ == "__main__":
    video_file = parse_args()
    run(video_file)
```

Βήμα 11: Εκτέλεση του προγράμματος
Εκτέλεση του προγράμματος
Εκτέλεση του προγράμματος
Για να εκτελέσετε τον code, γράψτε απλώς 'python3 file-name', όπου file-name είναι το όνομα του αρχείου σας. Στη δική μου περίπτωση είναι main.py. Η ανάλυση που θα δείτε θα πρέπει να μοιάζει με την επισυναπτόμενη εικόνα.

Κύρια σημεία που πρέπει να προσέξετε:

0.XY - Αυτή η τιμή, όπου XY είναι ακέραιος αριθμός, είναι το ποσοστό confidence που έχει το model για την ανίχνευση ενός ατόμου.
Σημειώστε ότι αυτό το script θα αποθηκεύσει επίσης τα δεδομένα σε CSV file, το

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

οποίο θα χρησιμοποιήσουμε στο στάδιο μετά τη συλλογή δεδομένων.

Βήμα 12: Ανάλυση μετά τη συλλογή δεδομένων
Ανάλυση μετά τη συλλογή δεδομένων
Αν χρησιμοποιήσατε τον τελευταίο ενδεικτικό code, θα λάβετε ένα γράφημα που δείχνει την ώρα της ημέρας σε σχέση με τον αριθμό των πτηνών που ανιχνεύτηκαν. Αυτό είναι ιδιαίτερα χρήσιμο για bird watchers, καθώς πολλά είδη ακολουθούν προβλέψιμα μοτίβα μετακίνησης σε σμήνη. Μελετώντας το γράφημα, οι παρατηρητές μπορούν να επιστρέψουν στην ίδια περιοχή σε παρόμοιες ώρες την επόμενη ημέρα, με μεγάλη πιθανότητα να εντοπίσουν ξανά τα ίδια πτηνά.

Βήμα 13: Επόμενα βήματα
Προς το παρόν, το έργο μου ανιχνεύει πότε υπάρχει πτηνό στο πεδίο παρατήρησης, αλλά σκοπεύω να αυξήσω την ακρίβειά του ώστε να αναγνωρίζει και το είδος του πτηνού. Για παράδειγμα, αν ο χρήστης θέλει να ειδοποιείται μόνο όταν εμφανίζεται περιστέρι, το model θα πρέπει να αγνοεί τα υπόλοιπα είδη πτηνών.

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

11. Κριτήρια αξιολόγησης:

- Επιτυχής ρύθμιση της camera
- Σωστή εκτέλεση του AI detection model
- Ακριβής καταγραφή των πτηνών που ανιχνεύονται
- Ικανότητα ερμηνείας των συλλεχθέντων δεδομένων

---

12. Διαθέσιμες γλώσσες:

Ελληνικά

---

13. Συντελεστές:

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

17. Ανατροφοδότηση και συνεχής βελτίωση:

Οι συμμετέχοντες ενθαρρύνονται να βελτιώσουν το model συλλέγοντας περισσότερες εικόνες πτηνών, εκπαιδεύοντας custom datasets ή προσθέτοντας αισθητήρες, όπως motion detectors.

Σημειώσεις υποβολής:

Βεβαιωθείτε ότι όλα τα απαιτούμενα έγγραφα και οι πόροι έχουν επισυναφθεί στην υποβολή σας. Η ομάδα αξιολόγησης του AI-SAVES@ProjectHub θα αξιολογήσει την υποβολή με βάση τις παρεχόμενες πληροφορίες. Μόλις εγκριθεί, το έργο θα δημοσιευθεί στο hub και οι εταίροι θα ενημερωθούν.

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.