**Title: 9-Month Daily Plan: UAV with CV, SLAM, 3D Mapping & Autonomy**

**🧠 Goal**: Build a UAV capable of autonomous navigation, real-time object detection/tracking, and 3D mapping using your existing computer vision/deep learning skills and developing new ones (SLAM, sensor fusion, robotics).

**⌛ Time per Day**: 5 hours **📆 Total Time**: ~1350 hours

| **Month** | **Focus Area** |
| --- | --- |
| 1 | Foundations: ROS2, Linux, Python, CV/ML Refresh |
| 2 | Object Detection, Tracking & Deployment |
| 3 | SLAM (Visual/ORB-SLAM2/OpenVSLAM) |
| 4 | UAV Basics: PID control, flight stack setup, manual control |
| 5 | Autonomy: Integrate SLAM + CV into UAV |
| 6 | Point Clouds & 3D Mapping (RTAB-Map, OctoMap, etc.) |
| 7 | Sensor Fusion: IMU + vision + depth |
| 8 | Final Integration + Field Testing |
| 9 | Polish: UI + Docs + Demo video + Portfolio writeup |

**Month 1: Foundations & ROS**

**Week 1: Linux & Python Refresher (You’ll go fast)**

**Goal**: Command line mastery, Python fluency for robotics/vision

* **Day 1-2**: Bash scripting, permissions, SSH, tmux, crontab, navigating logs
* **Day 3**: Git CLI + GitHub (SSH key, push/pull, branches, PR)
* **Day 4-5**: Python refresh: Classes, NumPy, file I/O, subprocess, decorators
* **Day 6**: Python threading & multiprocessing (for CV pipelines)
* **Day 7**: Mini project: CLI tool to batch process images with filters + logging

Week 2: ROS2 Basics (Humble preferred) **Goal: Publish/subscribe, parameters, TF2, rosbag2, RViz2**

* Day 8: ROS2 install + workspace + colcon + your first publisher
* Day 9: Subscriber + publisher with OpenCV camera feed
* Day 10: Launch files, ros2 launch, parameters
* Day 11: TF2 and static transforms
* Day 12: rosbag2 record/replay, rqt\_graph, RViz2 tools
* Day 13-14: Mini project: Webcam tracker in ROS2 node + visualization in RViz2

Week 3: Computer Vision Refresher (Fast Review)  
Goal: You know OpenCV, so focus on use in pipelines

* Day 15: Filtering, edge detection, contours
* Day 16: Background subtraction, motion detection
* Day 17: Color filtering, morphological ops, bitwise ops
* Day 18: Perspective transforms, stereo vision intro
* Day 19-20: Mini project: ROS2 node that detects and tracks color objects
* Day 21: Record rosbag2, visualize detections in RViz2

Week 4: DL Refresher (Fast Track)  
Goal: CNNs, PyTorch, real-time object detection

* Day 22: PyTorch refresh: Tensors, modules, training loop
* Day 23: Transfer learning with ResNet on small dataset
* Day 24-25: YOLOv5/YOLOv8 with your own dataset (or Roboflow)
* Day 26: Export YOLOv5 to ONNX/TorchScript
* Day 27: Integrate YOLO inference in ROS2
* Day 28: Mini project: Object detection ROS2 node using webcam

Month 2: Object Detection & Tracking in ROS2

Week 5: Tracking Algorithms  
Goal: Learn basic and DL-based trackers

* Day 29-30: OpenCV trackers (KCF, CSRT, MOSSE)
* Day 31: Deep SORT theory
* Day 32-33: Integrate YOLOv5 + Deep SORT
* Day 34-35: Mini project: Multi-object tracker in ROS2 + color coded IDs

Week 6: Metrics, Logging, Optimization  
Goal: Improve runtime and add diagnostics to CV pipelines

* Day 36: FPS measurement, latency bottlenecks
* Day 37: Use cv\_bridge, optimize ROS2 + OpenCV pipeline
* Day 38-39: Save cropped images of detected objects (dataset collector)
* Day 40-42: Mini project: Dataset collector + real-time tracker logger

Week 7: UAV Control Basics (DroneKit or PX4 SITL)  
Goal: Learn UAV control stack, simulation, and manual navigation

* Day 43-44: Learn MAVLink and PX4
* Day 45: Simulate UAV in Gazebo
* Day 46: Waypoint navigation
* Day 47-49: Mini project: Launch a simulated drone and follow path

Week 8: Object-Aware Navigation  
Goal: Combine perception with basic navigation behavior

* Day 50: Fuse detection and location from SLAM
* Day 51-54: UAV avoids or approaches objects based on detection
* Day 55-56: Mini project: Autonomous indoor navigation and object following

Month 3: SLAM and Mapping Fundamentals

Week 9: Visual and Lidar SLAM  
Goal: Learn and apply SLAM methods using camera and LIDAR in ROS2

* Day 57: SLAM concepts (pose graph, loop closure, map optimization)
* Day 58-59: Implement ORB-SLAM2 or OpenVSLAM in ROS2 with RGB-D data
* Day 60-61: Test SLAM setup and visualize 2D/3D maps in RViz2
* Day 62-63: Mini project: Run ORB-SLAM2 or OpenVSLAM in a simulated environment and visualize the generated map

Week 10: Lidar-based SLAM  
Goal: Use ROS2-compatible LIDAR-based SLAM systems

* Day 64: Hector SLAM and Cartographer overview
* Day 65-66: Integrate LIDAR with ROS2 and test Hector SLAM or Cartographer
* Day 67-68: Visualize 2D or 3D maps from Hector SLAM or Cartographer in RViz2
* Day 69-70: Mini project: Create a map with LIDAR data using Hector SLAM or Cartographer

Week 11: 3D Reconstruction and Mapping  
Goal: Generate and process 3D data from sensors

* Day 71: Point Cloud Library (PCL) basics: filtering, segmentation
* Day 72-73: Process point clouds using PCL in ROS2, visualize in RViz2
* Day 74-75: Learn Open3D basics and implement it for 3D mapping
* Day 76-77: Mini project: Build a 3D map using RGB-D data and Open3D

Week 12: Advanced Mapping Techniques  
Goal: Create usable spatial maps and integrate with movement

* Day 78: Occupancy grid mapping fundamentals
* Day 79-80: Implement 3D occupancy grid mapping with Open3D or Octomap
* Day 81-82: Integrate 3D mapping with robot movement in a simulated environment
* Day 83-84: Mini project: Create a 3D map while moving a robot in the simulated environment

**Month 4: UAV Platform Setup**

**Week 13: PX4 Autopilot & Drone Simulation**

* **Day 85**: Install PX4 autopilot and QGroundControl for UAV simulation.
* **Day 86-87**: Study UAV firmware and basic flight modes (Takeoff, hover, land).
* **Day 88**: Run a basic autonomous mission in Gazebo using PX4.
* **Day 89-90**: Mini Project: Simulate autonomous UAV flight with basic commands.

**Week 14: MAVROS Integration and Control**

* **Day 91**: Install and configure MAVROS for ROS2 integration.
* **Day 92-93**: Test communication between PX4 and ROS2 via MAVROS.
* **Day 94**: Implement simple UAV control using ROS2 with MAVROS (takeoff, hover, land).
* **Day 95-96**: Mini Project: Control a UAV in Gazebo via ROS2 using MAVROS and simple control commands.

**Week 15: UAV Dynamics and Control**

* **Day 97**: Study UAV dynamics: PID controllers, attitude control, altitude hold.
* **Day 98-99**: Implement basic UAV control with ROS2 (altitude and orientation).
* **Day 100**: Tune PID controllers for stable flight.
* **Day 101-102**: Mini Project: Simulate stable flight control in Gazebo with PID controllers.

**Week 16: Sensor Integration and UAV Testing**

* **Day 103**: Learn UAV sensor integration: IMU, depth camera, GPS.
* **Day 104-105**: Set up IMU sensor data and visualize in ROS2.
* **Day 106**: Integrate depth camera (RGB-D or stereo) with UAV and test data in ROS2.
* **Day 107-108**: Mini Project: Test UAV with integrated sensors (IMU, depth camera) in Gazebo, and visualize the data.

**Month 5: Object Tracking, Optical Flow, Kalman Filter**

**Week 17: Multi-Object Tracking + Optical Flow**

* **Day 109**: Study Deep SORT and ByteTrack algorithms for object tracking.
* **Day 110-111**: Implement Deep SORT or ByteTrack in ROS2 with video feed.
* **Day 112**: Learn Optical Flow methods (Lucas-Kanade, Farneback) in OpenCV.
* **Day 113-114**: Implement optical flow for tracking moving objects.
* **Day 115-116**: Mini Project: Track multiple objects in a video feed using Deep SORT/ByteTrack with optical flow in ROS2.

**Week 18: Kalman Filters & Sensor Fusion**

* **Day 117**: Study the basics of Kalman Filters (KF, EKF).
* **Day 118-119**: Implement a basic Kalman Filter for position tracking in Python.
* **Day 120**: Integrate IMU and GPS data for sensor fusion with Kalman Filters.
* **Day 121-122**: Apply Kalman Filter to fuse GPS + IMU data for position estimation in ROS2.
* **Day 123-124**: Mini Project: Position tracking with noisy sensor fusion (IMU + GPS) using a Kalman Filter in ROS2.

**Week 19: Advanced Sensor Fusion and Tracking**

* **Day 125**: Study advanced sensor fusion techniques (extended Kalman Filter, particle filter).
* **Day 126-127**: Implement Extended Kalman Filter (EKF) for UAV position tracking.
* **Day 128**: Explore tracking with visual features (optical flow, depth) combined with Kalman Filter.
* **Day 129-130**: Apply EKF and feature-based tracking for sensor fusion and real-time tracking in ROS2.
* **Day 131-132**: Mini Project: Implement visual feature-based tracking (depth + optical flow) with Kalman Filter for UAV navigation.

**Week 20: Object Avoidance and Collision Detection**

* **Day 133**: Study collision detection algorithms for UAVs (3D occupancy grid, proximity sensors).
* **Day 134-135**: Implement basic proximity detection using RGB-D or LIDAR data in ROS2.
* **Day 136**: Learn path planning algorithms for obstacle avoidance (RRT, A\*).
* **Day 137-138**: Implement basic obstacle avoidance in ROS2 with path planning algorithms.
* **Day 139-140**: Mini Project: Test and implement obstacle avoidance for a UAV in a simulated environment.

**Month 6: Advanced Mapping and Autonomy**

**Week 21: 3D Path Planning**

* **Day 141**: Study 3D path planning techniques (RRT, A\*).
* **Day 142-143**: Implement 3D path planning with MoveIt2 and ROS2.
* **Day 144**: Learn about Octomap for 3D occupancy grid mapping.
* **Day 145-146**: Integrate Octomap with path planning for 3D map-based navigation.
* **Day 147-148**: Mini Project: Implement and test path planning in a cluttered 3D environment using ROS2 and Octomap.

**Week 22: Path Following and Navigation**

* **Day 149**: Study path following controllers (Pure Pursuit, Stanley method).
* **Day 150-151**: Implement basic path following controller in ROS2 for UAV.
* **Day 152**: Integrate path following with MoveIt2 for autonomous flight.
* **Day 153-154**: Test path following in Gazebo with a simulated UAV.
* **Day 155-156**: Mini Project: Navigate a UAV along a predefined path in Gazebo using path following.

**Week 23: Autonomous Flight in Sim**

* **Day 157**: Learn advanced motion planning and trajectory generation for autonomous UAVs.
* **Day 158-159**: Implement motion planning algorithms for UAV navigation in Gazebo.
* **Day 160**: Integrate obstacle avoidance with path planning in a real-time simulation.
* **Day 161-162**: Test autonomous flight with real-time obstacle avoidance and navigation in Gazebo.
* **Day 163-164**: Mini Project: Perform obstacle avoidance + autonomous flight in Gazebo, including 3D mapping.

**Week 24: Simulation-based Real-world Testing**

* **Day 165**: Analyze UAV behaviors in various environments (indoor vs outdoor).
* **Day 166-167**: Test UAV in indoor simulation with dynamic obstacles (e.g., moving people or objects).
* **Day 168**: Implement real-time dynamic obstacle avoidance during navigation.
* **Day 169-170**: Test the UAV navigating in a simulated dynamic environment.
* **Day 171-172**: Mini Project: Autonomous navigation in a dynamic environment with real-time obstacle avoidance and mapping.

**Month 7: Real UAV Bring-Up**

**Week 25: Flight Test and Tuning**

* **Day 173**: Study the UAV hardware (motors, ESCs, propellers, etc.) and its calibration process.
* **Day 174**: Flash PX4 firmware onto the flight controller.
* **Day 175-176**: Connect UAV to ROS2 and MAVROS, test communication.
* **Day 177**: Perform ESC calibration and verify motor functionality.
* **Day 178-179**: Conduct a first real test: Autonomous takeoff and land in open space.
* **Day 180**: Mini Project: Test autonomous takeoff and land in open space with UAV using ROS2.

**Week 26: Basic Flight Control and Stability**

* **Day 181**: Study UAV control basics (PID, LQR).
* **Day 182-183**: Implement basic PID control for altitude and orientation in ROS2.
* **Day 184**: Test basic control (hover, yaw control) in a safe environment.
* **Day 185-186**: Fine-tune PID parameters for stable flight.
* **Day 187-188**: Mini Project: Test hover stability and basic flight control for pitch, roll, and yaw with UAV.

**Week 27: Real-World Mapping + CV**

* **Day 189**: Mount RGB-D camera on UAV.
* **Day 190-191**: Integrate camera feed with ROS2 for image processing and visualization in RViz.
* **Day 192**: Implement mapping algorithms (e.g., RTAB-Map, ORB-SLAM2) for real-time 3D mapping.
* **Day 193-194**: Test real-time point cloud generation and map building during flight.
* **Day 195-196**: Mini Project: Fly UAV indoors and build a 3D map using RGB-D camera.

**Week 28: GPS & Localization in the Real World**

* **Day 197**: Study GPS-based localization techniques for UAVs.
* **Day 198**: Integrate GPS with ROS2 for localization and path planning.
* **Day 199**: Test GPS-based navigation in open outdoor spaces with the UAV.
* **Day 200-201**: Fine-tune GPS and IMU fusion for accurate localization.
* **Day 202-203**: Mini Project: Test UAV localization in outdoor environment using GPS and IMU data.

**Month 8: Final System Integration**

**Week 29: Mission Planning + Full Autonomy**

* **Day 204**: Study mission planning frameworks for UAVs (e.g., QGroundControl, ROS2-based planner).
* **Day 205-206**: Set up waypoints and implement mission planner using ROS2 and MAVROS.
* **Day 207**: Test UAV flight with predefined mission and waypoint navigation.
* **Day 208-209**: Implement conditional behaviors for missions (e.g., stop for obstacle detection, follow moving targets).
* **Day 210-211**: Mini Project: Set up autonomous mission with predefined waypoints and object-following behavior in an outdoor environment.

**Week 30: Advanced Safety and Fault Tolerance**

* **Day 212**: Study fail-safe and fault tolerance strategies for UAV systems (e.g., GPS failure, battery low).
* **Day 213-214**: Implement return-to-home behavior when GPS signal is lost or battery is low.
* **Day 215**: Add redundant sensor fusion for more reliable operation (IMU + GPS + LIDAR).
* **Day 216-217**: Test safety behaviors and fail-safe mechanisms in real-world flight.
* **Day 218-219**: Mini Project: Implement and test fail-safe scenarios (e.g., return-to-home) in real-world flight.

**Week 31: System Refinement and Robustness**

* **Day 220**: Evaluate the overall performance of UAV system in dynamic environments.
* **Day 221-222**: Refine object detection, obstacle avoidance, and path planning in various real-world scenarios.
* **Day 223**: Fine-tune the flight control system (PID) for smoother, more stable flight in dynamic conditions.
* **Day 224-225**: Test the UAV's ability to follow moving targets and navigate cluttered environments with dynamic obstacles.
* **Day 226-227**: Mini Project: Test and refine autonomous flight in a dynamic and unpredictable environment (e.g., flying in a room with people or moving objects).

**Week 32: Final Demo Preparation**

* **Day 228**: Finalize the integration of all subsystems (detection, mapping, autonomy).
* **Day 229**: Record and document a full demonstration of the UAV's capabilities (mapping, tracking, and autonomy).
* **Day 230-231**: Create a detailed project report and presentation slides for your demonstration.
* **Day 232**: Record video for portfolio and YouTube showcasing your UAV's capabilities.
* **Day 233-234**: Mini Project: Conduct a final demonstration of the fully autonomous UAV, showing mapping, tracking, and navigation.

**Month 9: Final Project + Portfolio Building**

**Week 33: Final Project Execution**

* **Day 235**: Final integration of the UAV system (visual SLAM, path planning, object tracking).
* **Day 236**: Test the UAV in complex environments (e.g., indoor/outdoor, different obstacles, moving targets).
* **Day 237-238**: Record a comprehensive demonstration of the system: mapping, object tracking, autonomous navigation.
* **Day 239**: Evaluate and troubleshoot any last-minute issues with real-world flight performance.
* **Day 240**: Document the system architecture, key ROS2 packages used, and workflow for your UAV project.
* **Day 241-242**: Mini Project: Conduct a final run of the UAV in a real environment with obstacles and moving targets while recording video.

**Week 34: Polish and Portfolio Creation**

* **Day 243**: Clean up your code and structure the repository for public release on GitHub.
* **Day 244**: Add detailed comments to the codebase and provide usage instructions.
* **Day 245**: Create README file that explains the entire project: hardware, software, and steps to run.
* **Day 246-247**: Create YouTube video showcasing the final demonstration (with commentary and technical details).
* **Day 248**: Review project for completeness: Is there any part you want to improve? Test again.
* **Day 249-250**: Mini Project: Polish and prepare your final demo video for online portfolio presentation.

**Week 35: Portfolio & Resume Building**

* **Day 251**: Update resume with detailed project experience and skills.
* **Day 252**: Add your project to your portfolio site (or GitHub for open-source contributions).
* **Day 253**: Prepare a short technical description of your project for job applications (1-2 paragraphs).
* **Day 254**: Add your YouTube video link to your resume/portfolio to demonstrate your practical skills.
* **Day 255-256**: Create a LinkedIn post showcasing your work and linking to your GitHub/YouTube.
* **Day 257-258**: Mini Project: Draft cover letter tailored to jobs in UAV/robotics, highlighting your skills and portfolio.

**Week 36: Job Applications and Final Evaluation**

* **Day 259**: Begin applying to jobs related to robotics, UAVs, and computer vision, with your portfolio ready.
* **Day 260-261**: Apply to positions in Japan related to your desired job field.
* **Day 262**: Prepare for interviews by reviewing key concepts, algorithms, and project specifics.
* **Day 263-264**: Practice presenting your UAV project to interview panels (focus on both technical details and high-level concepts).
* **Day 265**: Mini Project: Send out 3-5 job applications, tailor your resume for each role.