Efficient Self-Supervised Neuro-Analytic Visual Servoing for Real-time Quadrotor Control

Sebastian Mocanu [1] Sebastian-Ion Nae [1] Mihai-Eugen Barbu [1] Marius Leordeanu [1, 2, 3]

[1]National University of Science and Technology POLITEHNICA Bucharest, Romania [2]Institute of Mathematics "Simion Stoilow" of the Romanian Academy, Romania [3]NORCE Norwegian Research Center, Norway

Motivation and Contribution

Challenge: Classical IBVS methods suffer from numerical instabilities and singularities, while marker-based approaches (ArUco, AprilTags) limit deployment in dynamic indoor environments. GPS-denied scenarios demand efficient, marker-free visual servoing for quadrotor control.

Our Solution: We present a self-supervised neuro-analytical framework featuring a **N**umerically **S**table **E**fficient and **R**educed (NSER) IBVS teacher model, distilled into a lightweight 1.7M parameter student network achieving **11×** real-time performance with **improved control accuracy**.

Key Contributions:

- **Stable analytical teacher:** Improved IBVS controller solving numerical instabilities through reduced classical equations, enabling robust marker-free control.
- **Two-stage segmentation:** YOLOv11 + U-Net mask splitter for anterior-posterior vehicle segmentation, accurately estimating target orientation.
- **Efficient knowledge distillation:** Dual-path system transferring geometric visual servoing from teacher to compact student neural network that *outperforms* the teacher while suitable for onboard deployment.
- **Practical sim-to-real transfer:** Digital-twin training with real-world fine-tuning, validated in GPS-denied indoor environments with minimal hardware.

Limitations in existing methods: The core challenge

Indoor quadrotor navigation faces critical challenges:

Classical IBVS Methods

- Numerical instabilities from singularities in interaction matrix
- Conditioning issues during large camera motions
- Suboptimal convergence in dynamic scenarios
- Require careful feature correspondence

Marker-Based Approaches

- Dependency on ArUco/AprilTag fiducial markers
- Sensitive to lighting conditions & occlusion
- Marker installation/maintenance overhead
- Limited deployment in unstructured environments

Open Challenge: How to achieve *marker-free*, *numerically stable*, *real-time* visual servoing in GPS-denied indoor environments with *minimal computational resources* for onboard deployment?

Requirements:

- ✓ **Stability:** Robust control without numerical instabilities
- ✓ **Marker-free:** No dependency on fiducial markers or external infrastructure
- ✓ **Real-time:** Suitable for onboard deployment on resource-constrained platforms
- ✓ **GPS-denied:** Vision-only navigation in indoor non-GPS environments
- ✓ **Cost-effective:** Minimal training data and hardware requirements

Our Solution: Framework Overview

Teacher-Student Knowledge Distillation Architecture

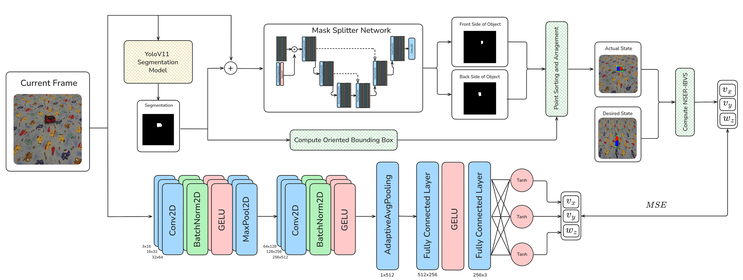


Figure 1. Overview of the proposed Teacher-Student Self-supervised Neuro-Analytic model. Top row illustrates the NSER-IBVS (analytical) Teacher path, which uses YOLO for object segmentation and a mask splitter network to estimate object orientation relative to the drone camera. The bottom one represents the Student path, which learns by minimizing the MSE cost between its output drone commands (v_x, v_y, ω_z) and the Teacher's output.

Teacher Path: NSER-IBVS

Analytical & Numerically Stable

1. **YOLOv11 Nano** (2.84M params)
   Object detection & segmentation
2. **U-Net Mask Splitter** (1.94M params)
   Anterior-posterior segmentation to compute orientation
3. **Stable Keypoint Extraction**
   Clockwise ordering from mask centroids
4. **Reduced IBVS Controller**
   Computes v_x, v_y, ω_z only

Role: Generate training labels
(velocity commands for student)

Student Path: Lightweight CNN

Compact & Real-Time

1. **Input:** RGB image (224×224)
   Given an single RGB image to compute velocities
2. **Convolutional Network Backbone**
   6 conv blocks (16→512 channels)
3. **Global Average Pooling**
   Spatial dimension reduction
4. **Fully-Connected Layers**
   Direct velocity regression (v_x, v_y, ω_z)

Role: End-to-end deployment
Only **1.7M params**. **11× faster**

Technical Contribution: Student learns *implicitly* from teacher without explicit supervision
Training: Minimize MSE between student predictions and teacher commands
Result: Student achieves similar (real-world environment) or better (digital-twin environment) control with **11× speedup**

Numerically Stable Efficient Reduced Image-Based Visual Servoing (NSER-IBVS)

Two-Stage Segmentation for Stable Feature Detection

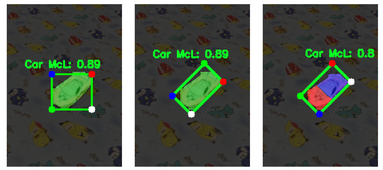


Figure 2. Comparison of different bounding box approaches derived from segmentation results: (**left**) regular bounding box including parts that do not belong to the object, (**middle**) oriented bounding box that may vary in orientation between frames, and (**right**) oriented bounding box using a mask-splitting network to separate anterior and posterior vehicle components for improved ordering stability.

1. **YOLOv11 Nano** (2.84M params)
   Detects vehicle & generates segmentation mask
2. **U-Net Mask Splitter** (1.94M params)
   Splits mask into front/back regions
   Uses attention mechanisms for precise division
3. **Oriented Bounding Box**
   Clockwise keypoint ordering from centroids
   4 stable corner points for IBVS

Reduced IBVS: 6-DOF → 3-DOF

Classical IBVS controls 6 velocities $(v_x, v_y, v_z, \omega_x, \omega_y, \omega_z)$, causing numerical instabilities. For fixed-altitude quadrotors, we reduce to 3 essential commands:

$$\begin{pmatrix} \dot{u} \\ \dot{v} \end{pmatrix} = \begin{pmatrix} -\frac{f}{Z} & 0 & \bar{v} \\ 0 & -\frac{f}{Z} & -\bar{u} \end{pmatrix} \begin{pmatrix} v_x \\ v_y \\ \omega_z \end{pmatrix} \quad (1)$$

Reducing IBVS from 6-DOF to 3-DOF matches the drone's physical constraints (the two angular velocities can't be controlled by our quadrotor and the vertical linear velocity is not needed in this setup) for faster, more stable visual servoing.

Benefits of this approach:

- Resolves singularities in interaction matrix
- Ensure stable feature correspondence across consecutive frames
- Enables marker-free visual servoing using four keypoints via the two-stage segmentation pipeline, following $\mathbf{v} = -\lambda \mathbf{L}_s^+ (\mathbf{s} - \mathbf{s}^*)$

Knowledge Distillation & Data Generation

Real-Time Student Network (1.7M params)

- Direct RGB → velocity regression with MSE loss
- Conv layers: 16→512 channels (batch norm + GELU)
- Target normalization for stable training
- **Two-stage:** Pretrain on sim, fine-tune on real data

Digital Twin & Dataset

- Sphinx/Unreal Engine 4 simulation (5×4m setup)
- **Sim:** 14,693 train / 1,123 val frames
- **Real:** 13,760 train / 1,084 val frames
- Multiple gimbal angles (30°-90°) for diversity
- Custom mask splitter tool for front/back labeling

Website & Demos

Lab Website

Experimental Setup

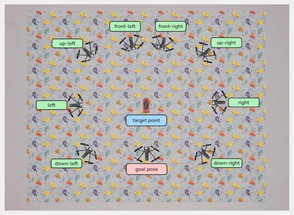
Dataset & Training Protocol

Simulation Evaluation:

- **800** total test runs
  (100 runs × 8 drone poses)
- Student trained on only **240** runs
  (30 per pose, 74,307 frames)
- Data augmentation: **5×** factor
  (brightness, saturation, noise)
- Input: 224×224 RGB images
- Output: normalized (v_x, v_y, ω_z)
- Optimizer: Adam (lr=0.001)
- Loss: MSE with early stopping

Real-World Validation:

- **80** flight sequences
  (10 per pose, 43,963 frames)
- Digital-twin pre-training
- Real-world fine-tuning
- Same termination conditions



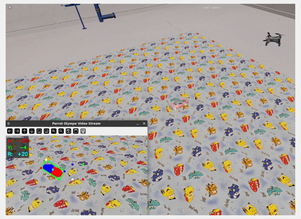


Figure 3. Left represents 8 starting poses: front positions with the drone oriented at ±14°, and corner positions with the drone is oriented at ±45°, the task is to navigate to the goal pose behind the car. Right represents an NSER-IBVS experiment in the digital-twin environment.

Termination Conditions & Evaluation Metrics

Teacher (NSER-IBVS):

1. Hard timeout: 75 seconds
2. Soft: median error ≤ 80 px + zero velocity (3s)
3. Hard: median error ≤ 40 px (3s)

Student (ConvNet):

- Absolute velocity ≤ 1 for 3 consecutive seconds
- Empirically derived from teacher statistics

Evaluation Metrics:

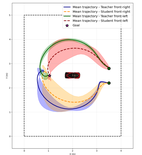
- **Error Norm:** L2 distance between current and goal keypoints (last 3s)
- **IoU:** Intersection over Union of bounding boxes
- **Flight Duration:** Total time from takeoff to landing
- **Flight Distance:** Total path length traveled
- **Inference Speed:** FPS comparison

Results

The results demonstrate that the student model achieves significantly improved tracking accuracy compared to the teacher method. In simulation, the student model achieves a mean error norm of 14.261 pixels compared to the teacher 29.756 pixels, representing a 52% improvement in tracking precision. The IoU metric shows similar improvements, with the student achieving 0.752 compared to the teacher 0.522, indicating better object coverage and localization in Table 1.

		Flight SIM (distance(m) / time(s)) ↓	Norm Error SIM (px) ↓	IoU SIM ↑	Flight (distance(m) / time(s)) ↓	Norm Error (px) ↓	IoU ↑
Up-Right	Teacher	**5.675 / 24.226**	31.800	0.503	**5.622 / 41.581**	31.499	0.621
	Student	6.064 / 28.298	**13.172**	**0.766**	5.716 / 45.885	**22.802**	**0.6919**
Front-Left	Teacher	6.196 / **27.315**	30.706	0.517	6.493 / **37.535**	**28.54**	**0.611**
	Student	**6.041** / 27.917	**13.430**	**0.758**	**6.490** / 47.238	33.981	0.560
Right	Teacher	**4.317 / 19.637**	31.137	0.494	4.831 / **41.409**	**32.423**	**0.612**
	Student	4.518 / 21.987	**13.798**	**0.759**	**4.811** / 57.245	43.672	0.5
Down-Left	Teacher	2.779 / 15.988	28.473	0.518	4.384 / **31.622**	**28.00**	**0.611**
	Student	**2.777 / 14.900**	**13.257**	**0.763**	**4.326** / 41.044	39.531	0.5253
Mean	Teacher	**4.774 / 22.111**	29.756	0.522	**5.253 / 36.185**	**29.956**	**0.627**
	Student	4.859 / 23.790	**14.261**	**0.752**	5.300 / 45.369	33.334	0.591

Table 1. Performance comparison; left side - results in the simulator | right side - results on real-world flights



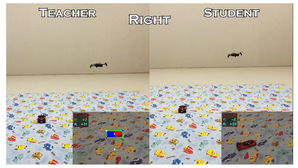


Figure 4. Trajectories of teacher and student simulation flights, with mean and standard deviation for 2 starting poses (green circles: front-left and front-right). The solid blue and green lines represent mean paths of teacher NSER IBVS method, while dashed orange and red lines show student paths. Shaded regions indicate trajectory variability across runs. The star represents the goal pose. Note that the student displays more path variation, but it has a shorter average path than the teacher. On the right are the visualizations of the real-world drone perspective and actor perspective.

The trajectory evolution analysis reveals that both methods successfully converge toward the target, but with different characteristics. The teacher method shows more conservative command profiles with gradual convergence, while the student method demonstrates more direct approaches with faster error reduction rates shown in Figure 4.

Evaluator	Avg	Std	Med	Min	Max	FPS
NSER IBVS	20.69	7.63	24.56	6.45	82.55	48.30
Student	**1.85**	0.93	1.84	1.79	235.64	**540.8**

Table 2. Computation times (ms). The small 1.7M params student ConvNet is 11× faster (540.8 FPS) than the teacher (48.3 FPS)

Student - key observations:

- Significantly better computation with **11× speedup**
- Robust **tracking** precision (err norm: **14.2** | **33.3**)
- Better **object coverage** (mIoU: **0.75** | **0.59**)
- Fine-tuning on real-world scenes proves effective for domain adaptation

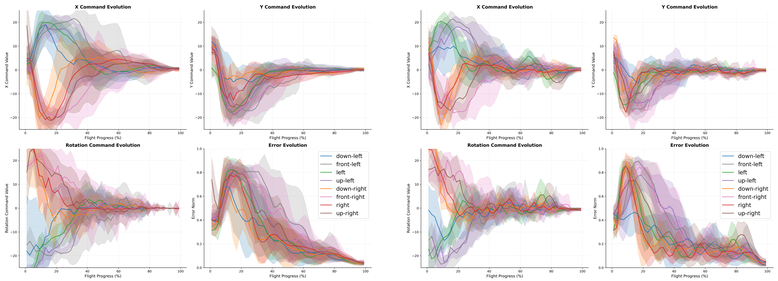


Figure 5. Comparison of drone control command and error evolution on novel test sequences for (a) teacher (NSER IBVS) and (b) student (Self-Supervised Neuro-Analytical) across flight trajectories from 8 different starting points (see Figure 3). Solid lines represent mean values; shaded areas indicate variability across runs. All control commands and errors converge toward zero, indicating robust trajectory tracking and control convergence from various initial conditions. Note the striking similarity in drone control and distance error to target between the complex analytical teacher and the small ConvNet student.

Conclusions

- First fully self-supervised, **marker-free** visual servoing framework for quadrotors.
- Efficient knowledge distillation: a lightweight student model ($1.7M$ parameters) achieves **11× faster inference** ($529.8FPS$) than the analytical teacher ($48.3FPS$) while maintaining comparable accuracy.
- Practical sim-to-real transfer requiring only **80** real-world flights for adaptation.
- Real-time performance suitable for **onboard deployment**.

Acknowledgements

This work is supported by projects:

- Romanian Hub for Artificial Intelligence - HRIA, Smart Growth, Digitization and Financial Instruments Program, 2021-2027 (MySMIS No. 334906)
- European Health and Digital Executive Agency (HADEA) through DIGITWIN4CIUE (Grant No. 101084054)
- European Lighthouse of AI for Sustainability - ELIAS, Horizon Europe program (Grant No. 101120237)