

Article

A Late-Fusion Multimodal Approach for Safety-Aware Workspace Modeling in Collaborative Robotic Systems

Kevin David Ortega-Quinones , Elias Escobar-Pereira , Michael Felipe Cifuentes-Molano ,
Germán Andrés Holguín-Londoño  and Mauricio Holguín-Londoño 

Electrical Engineering Program, Faculty of Engineering, Technological University of Pereira,
Pereira 660003, Colombia; elias.escobar@utp.edu.co (E.E.-P.); mcifuentes@utp.edu.co (M.F.C.-M);
gahol@utp.edu.co (G.A.H.-L.); mau.hol@utp.edu.co (M.H.-L.)

* Correspondence: kevin.ortega@utp.edu.co

Abstract

Ensuring safe coexistence between human operators and industrial robot manipulators is a critical challenge in collaborative manufacturing environments. Existing approaches rely either on dedicated safety-rated hardware, which is expensive and difficult to retrofit, or on purely vision-based classifiers that discard the precise kinematic state available from the robot controller, leading to unresolved visual ambiguities when different joint configurations produce similar appearances from fixed camera viewpoints. Kinematics-only approaches, while precise, lack the spatial context needed to disambiguate configurations near workspace boundaries. We propose RGBJointsNet, a late-fusion multimodal deep learning classifier that combines RGB visual features extracted by a frozen EfficientNet-B2 convolutional backbone with a compact kinematic stream processing the 12-dimensional joint angle vector of a dual-UR5 robotic cell. The model maps each observation to one of five mutually exclusive workspace zones: rest (C_0), nominal (C_1), extended (C_2), shared/collision-risk (C_3), and joint-limit/singularity (C_4). A dedicated simulation environment built on ROS 2 Humble Hawksbill and Gazebo Classic 11 was used to generate a labelled dataset of 54,309 frames and 162,927 RGB images from three calibrated overhead cameras, with analytic ground-truth labels derived from closed-form forward kinematics. Training on a CPU with a feature-caching strategy brings the per-epoch wall-clock time to seconds, making the approach tractable without GPU hardware. On the held-out test set, the model achieves 87.1% overall accuracy and a macro-averaged F1 score of 90.0%, with near-perfect recall of 99.3% for the safety-critical shared zone C_3 . The trained classifier is integrated as an ROS 2 inference node capable of running at 10 Hz on a standard workstation. Our results demonstrate that joint angle information is a decisive complement to RGB imagery for fine-grained, safety-oriented workspace classification in simulation-derived settings.

Keywords: collaborative robotics; workspace zone classification; multimodal deep learning; RGBJointsNet; EfficientNet; ROS 2; Gazebo; sim-to-real; robot safety; UR5



Academic Editor: Beno Benhabib

Received: 15 April 2026

Revised: 22 May 2026

Accepted: 29 May 2026

Published: 30 June 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

The integration of multiple robot manipulators into shared workspaces is a growing trend in collaborative manufacturing, where achieving high productivity while guaranteeing safety remains a core challenge [1,2]. When two or more industrial arms operate in overlapping reachable volumes, real-time awareness of their mutual workspace state is essential both for preventing robot–robot collisions and for ensuring the safety of human operators who may coexist in the same space.

Industrial safety standards such as ISO 10218-1 [3] and ISO 10218-2 [4] formalise the requirements for speed and separation monitoring, hand-guiding, and power and force limiting; however, their implementation typically relies on dedicated safety-rated hardware, such as laser scanners, safety PLCs, and force/torque sensors, which are both expensive and difficult to retrofit into existing production lines.

In multi-robot cells, classifying the inter-robot workspace state constitutes a necessary first layer of any safety architecture: a supervisory system must assess whether the robots are in a hazardous mutual configuration before evaluating the risk posed to a nearby human operator.

A complementary and cost-effective approach is to use the information already available in modern robot cell joint encoders and overhead RGB-D cameras to infer the spatial and kinematic state of the system in real time. When two robot arms share a common workspace, five qualitatively distinct operational zones can be identified analytically from the joint state: a rest configuration (C_0), a nominal operating region (C_1), an extended-reach region (C_2), a collision-risk shared zone (C_3), and a joint-limit or kinematic-singularity boundary (C_4). Early detection of transitions into the shared zone (C_3) or the limit zone (C_4) enables supervisory controllers to trigger speed reductions or emergency stops before a hazardous condition develops [5].

Deep learning has fundamentally changed the landscape of robot perception, with convolutional neural networks achieving superhuman performance on visual recognition tasks and enabling robots to interpret rich scene information from standard cameras [6]. Transfer learning from large-scale image datasets such as ImageNet further reduces the labelled-data requirements, making fine-grained classification feasible with moderate-sized domain-specific datasets [7]. Nevertheless, purely vision-based classifiers discard the precise kinematic information available from the robot controller, which encodes the exact configuration of each joint and is essential for resolving visual ambiguities that arise when different configurations produce similar appearances from fixed camera viewpoints.

Multimodal deep learning [8] addresses this limitation by jointly learning from heterogeneous sensor streams. Late-fusion architectures, in which each modality is processed by a dedicated subnetwork before combining their representations in a shared head, have been shown to be particularly effective when the input modalities have fundamentally different statistical structures [9]. In the present context, the RGB stream provides global scene context, robot pose, link occlusions, and inter-robot proximity while the kinematic stream provides precise, noise-free measurement of the robot state; these two signals are complementary by design.

Beyond collaborative robotics, multimodal deep learning has demonstrated significant advances in related domains. In supervised settings, reversible guidance and cyclic knowledge distillation have been used to combine heterogeneous modalities for single-image super-resolution reconstruction [10]. In unsupervised fault diagnosis, collaborative spatial-channel graph interaction with multi-scale transfer learning enables cross-domain feature alignment without labelled target data [11]. These works illustrate the broader trend of exploiting complementary modality interactions for robust feature extraction, a principle that directly motivates the late-fusion design of the present work.

The main contribution of this work lies in the integration of visual and kinematic information for the fine-grained classification of safety zones in dual-arm robotic systems. Unlike approaches based solely on images or predefined rules, the proposed methodology combines RGB images with joint angles, taking advantage of the complementary strengths of both sources of information: RGB vision provides global scene context, occlusion information, and inter-robot proximity cues, whereas joint variables accurately describe the kinematic state of the system without depending on the camera viewpoint.

The zone taxonomy is grounded in formal kinematic definitions based on forward kinematics, the Yoshikawa manipulability index, and the reach radius, enabling a consistent, interpretable, and reproducible classification. Furthermore, the labelling process is performed through an analytical function derived from forward kinematics, avoiding the subjectivity of manual annotation and enabling deterministic reproduction—a property particularly relevant in industrial applications where traceability and reliability are fundamental requirements.

A natural question is why a learned multimodal classifier is preferred over the direct deployment of the analytic labelling function $f: \mathbb{R}^{12} \rightarrow \{C_0, \dots, C_4\}$. While this function can compute the zone exactly from joint angles in the controlled simulation environment, the inclusion of the visual modality is motivated by three considerations. First, sensor redundancy is a safety design principle: standards such as ISO 10218 [3,4] require that safety-critical functions do not depend on a single sensing channel; the dual-stream architecture enables cross-validation between modalities, so that a disagreement between the kinematic and visual predictions triggers a conservative safety response. Second, the visual stream captures workspace information not encoded in the joint vector: inter-link clearances arising from different elbow configurations, environmental context (floor, obstacles, mounting structures), and, in planned future extensions, the presence and pose of human operators. Third, the learned classifier generalises to deployment scenarios where the analytic function may be unavailable or incomplete, such as legacy manipulators with uncertain kinematic parameters or uncalibrated encoders, or workspaces containing unmodelled objects.

A further practical challenge is the cost of annotated data. Physical experiments with two industrial manipulators are time-consuming, require safety measures, and cannot easily reproduce hazardous configurations such as near-collisions. Photo-realistic robot simulators offer an attractive alternative: configurations can be sampled exhaustively, analytically labelled, and reproduced deterministically without risk. Recent surveys confirm that simulation-to-real transfer is reliable for appearance-based tasks when the simulator accurately models the robot geometry and camera parameters [12,13]. Moreover, the ROS 2 [14] middleware enables seamless migration of a simulation-trained inference module to physical hardware by replacing simulated topics with hardware interface abstractions.

This paper makes the following contributions:

1. A publicly reproducible simulation pipeline using ROS 2 Humble and Gazebo Classic 11 for generating large, analytically labelled datasets of dual-arm robot configurations, including rare and safety-critical events.
2. A rigorous five-class kinematic taxonomy with formal closed-form definitions based on reach radius, inter-effector distance, and Yoshikawa manipulability, enabling analytic ground-truth labelling without human annotation.
3. RGBJointsNet, a late-fusion multimodal classifier that explicitly exploits the complementarity of view-invariant kinematic features (12-D joint angles) and appearance features (RGB imagery); the architecture is trainable on a CPU in minutes via a feature-caching strategy, removing GPU hardware dependency.
4. Quantitative evaluation on a 54,309-frame dataset showing 87.1% overall accuracy and 99.3% recall on the safety-critical shared zone.
5. A deployable ROS 2 inference node operating at 10 Hz with a standard eight-core CPU.

The remainder of the paper is organised as follows. Section 2 describes the simulation environment, zone taxonomy, dataset construction, camera calibration, skeleton reprojection, model architecture, and training protocol. Section 3 presents the experimental

results and discusses their implications. Section 4 summarises the findings and outlines future work.

2. Materials and Methods

This section describes the complete experimental pipeline, from the design of the robotic simulation environment and the construction of the labelled dataset to the camera calibration procedure, the skeleton reprojection method, and the architecture and training protocol of the proposed multimodal classifier. Figure 1 summarises the end-to-end pipeline.

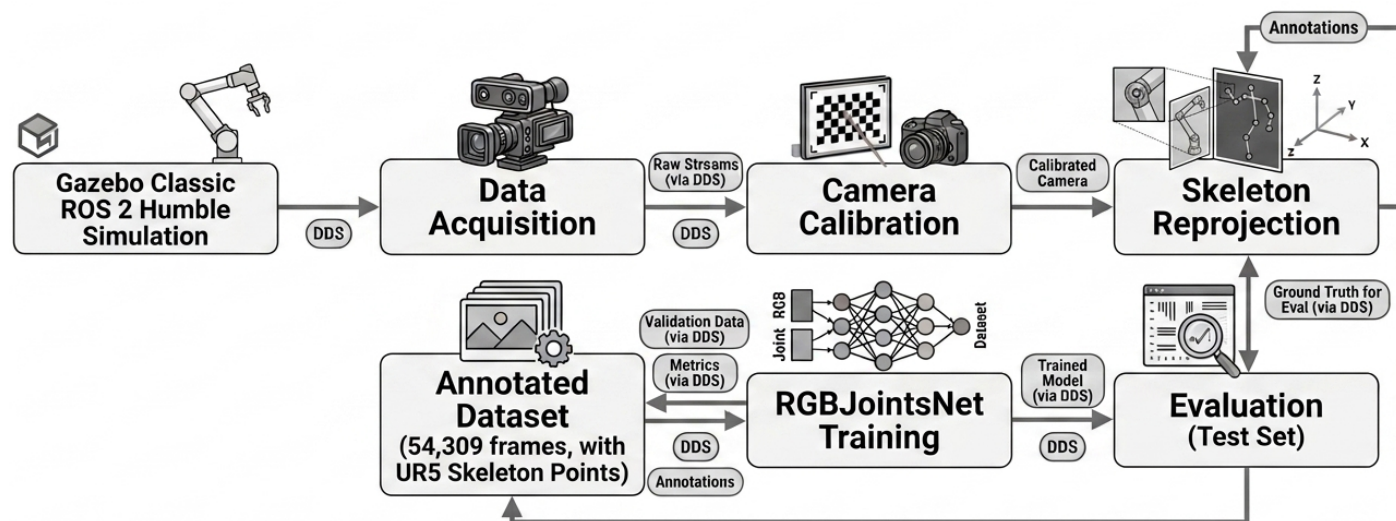


Figure 1. Data flow pipeline of the proposed system. Joint states ($\mathbf{q} \in \mathbb{R}^{12}$) and RGB images from three calibrated cameras are acquired from the simulated dual-UR5 cell, processed through the analytic labelling function and frozen-backbone feature extraction, and fused by the RGBJointsNet classifier into a five-class zone prediction. The trained model is deployed as an ROS 2 inference node communicating via DDS.

2.1. Simulation Environment

2.1.1. Rationale for ROS 2 and Gazebo

The experiment was conducted entirely in simulation for three reasons: (i) repeatability: every configuration is deterministically recoverable from the stored joint vector; (ii) safety: hazardous class C3 (shared zone) and class C4 (workspace limits) can be generated without risk; (iii) ground-truth availability: the analytic labelling function $f : \mathbb{R}^{12} \rightarrow \{C_0, \dots, C_4\}$ (Section 2.2) yields noise-free labels from the simulator's internal state.

ROS 2 Humble Hawksbill [14] was selected as the middleware because (i) it is the current Long-Term Support (LTS) release, (ii) it provides real-time-capable deterministic scheduling via DDS, and (iii) its `ros2_control` framework [15] enables seamless migration of the trained classifier to physical hardware by replacing simulation topics with hardware interface abstractions.

Gazebo Classic 11 [16] was chosen over newer Ignition/Fortress releases because the `gazebo_ros2_control` plugin for Humble had more community support and validated UR5 models at the time of development. The physics engine (ODE, step size 1 ms) accurately models the UR5 dynamics under the rigid-body assumption sufficient for appearance-based zone classification.

2.1.2. Robotic Cell Layout

The simulated cell comprises two Universal Robots UR5 manipulators [17] and three fixed RGB-D cameras. The UR5 is a 6-DoF serial manipulator with a maximum reach of 850 mm and a payload of 5 kg. Its Denavit–Hartenberg parameters [18] are listed in Table 1.

Table 1. Modified Denavit–Hartenberg parameters of the UR5 manipulator. Joint ranges are $[-2\pi, 2\pi]$ rad for all joints.

Joint i	a_i (m)	d_i (m)	α_i (rad)	$\theta_{i,0}$ (rad)
1	0.00000	0.089159	$\pi/2$	0
2	−0.42500	0.000000	0	0
3	−0.39225	0.000000	0	0
4	0.00000	0.109150	$\pi/2$	0
5	0.00000	0.094650	$-\pi/2$	0
6	0.00000	0.082300	0	0

The robots are positioned symmetrically about the world origin at height $z_b = 1.305$ m (mounting pedestal):

$$\mathbf{t}_1 = (0, +0.9, 1.305)^\top \text{ m}, \quad \psi_1 = 0, \quad (1)$$

$$\mathbf{t}_2 = (0, -0.9, 1.305)^\top \text{ m}, \quad \psi_2 = \pi, \quad (2)$$

where ψ_i is the yaw angle of robot i in the world frame. This symmetric arrangement maximises the overlap of the two reachable workspaces, increasing the frequency of safety-critical class-C3 events.

Three RGB-D cameras are placed at the vertices of an equilateral triangle of radius $r_c = 2.5$ m at height $z_c = 3.5$ m, each pointing toward the workspace centroid $\mathbf{c} = (0, 0, 1.4)^\top$ m (Table 2).

Table 2. Camera positions and orientations in the Gazebo world frame. $\phi_c = \arctan((z_c - c_z)/r_c) \approx 0.698$ rad is the fixed depression angle.

Camera	x (m)	y (m)	z (m)	Pitch (rad)	Yaw (rad)
cam1	+2.500	0.000	3.5	0.6981	π
cam2	−1.250	+2.165	3.5	0.6981	$5\pi/3$
cam3	−1.250	−2.165	3.5	0.6981	$\pi/3$

The simulated robotic cell is shown in Figure 2. The two UR5 manipulators operate in a shared workspace monitored by a three-dimensional perception system, whose geometric arrangement ensures full visual coverage of the interaction zones between both arms.

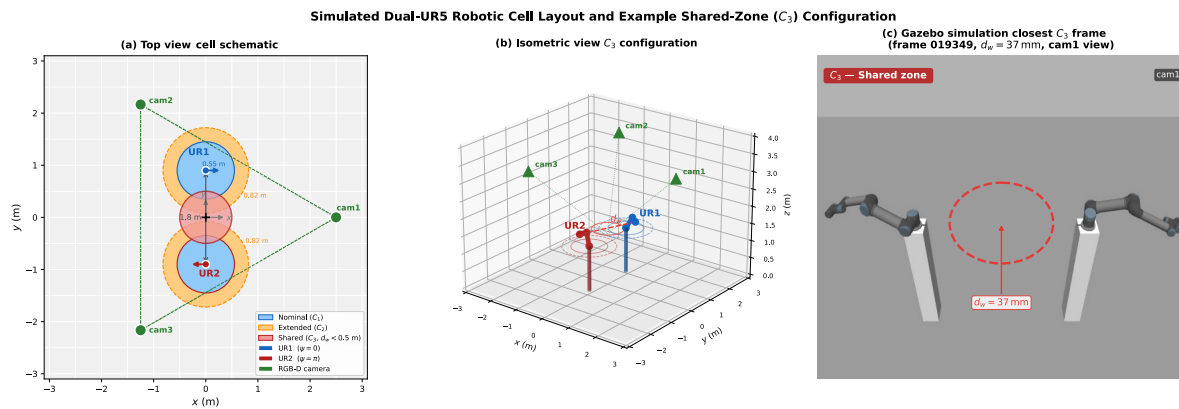


Figure 2. Simulated robotic cell. (a) Top view: the two UR5 manipulators (UR1, UR2) are mounted symmetrically at $y = \pm 0.9$ m. The three RGB-D cameras (blue dots) form an equilateral triangle of radius $r_c = 2.5$ m. (b) Isometric view from Gazebo showing a class- C_3 (shared-zone) configuration. (c) Real Gazebo image.

2.2. Zone Classification Taxonomy

Five mutually exclusive workspace classes are defined (Table 3) based on kinematic quantities computable in closed form from the joint vector $\mathbf{q} = (q_1^{(1)}, \dots, q_6^{(1)}, q_1^{(2)}, \dots, q_6^{(2)})^\top \in \mathbb{R}^{12}$.

Reach radius.

The planar reach radius of robot i is

$$r_i(\mathbf{q}^{(i)}) = \sqrt{p_x(\mathbf{q}^{(i)})^2 + p_y(\mathbf{q}^{(i)})^2}, \quad (3)$$

where $\mathbf{p}(\mathbf{q}^{(i)}) \in \mathbb{R}^3$ is the end-effector position in the robot's `base_link` frame, obtained via forward kinematics (FK).

Forward kinematics.

The homogeneous transformation from the base frame of robot i to the end-effector is

$${}^0T_6 = \prod_{k=1}^6 {}^{k-1}T_k(q_k), \quad (4)$$

where the elementary DH matrix is

$${}^{k-1}T_k = \begin{pmatrix} c_k & -s_k c_{\alpha_k} & s_k s_{\alpha_k} & a_k c_k \\ s_k & c_k c_{\alpha_k} & -c_k s_{\alpha_k} & a_k s_k \\ 0 & s_{\alpha_k} & c_{\alpha_k} & d_k \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (5)$$

with $c_k = \cos q_k$, $s_k = \sin q_k$, $c_{\alpha_k} = \cos \alpha_k$, $s_{\alpha_k} = \sin \alpha_k$.

World-frame end-effector position.

The end-effector of robot i in the world frame is

$$\mathbf{p}_{e,i}^w = \mathbf{t}_i + \mathbf{R}_z(\psi_i) \mathbf{p}(\mathbf{q}^{(i)}), \quad (6)$$

where $\mathbf{R}_z(\psi_i) \in SO(3)$ is the rotation matrix corresponding to the robot's base yaw angle ψ_i .

Inter-effector distance.

$$d_w = \|\mathbf{p}_{e,1}^w - \mathbf{p}_{e,2}^w\|_2. \quad (7)$$

Yoshikawa manipulability.

Proximity to kinematic singularities is quantified by the Yoshikawa manipulability index [19]:

$$w(\mathbf{q}) = \sqrt{\det(\mathbf{J}_p(\mathbf{q}) \mathbf{J}_p(\mathbf{q})^\top)}, \quad (8)$$

where $\mathbf{J}_p \in \mathbb{R}^{3 \times 6}$ is the positional sub-Jacobian, estimated numerically:

$$[\mathbf{J}_p]_{:,k} = \frac{\mathbf{p}(\mathbf{q} + \varepsilon \mathbf{e}_k) - \mathbf{p}(\mathbf{q})}{\varepsilon}, \quad \varepsilon = 10^{-5} \text{ rad}, \quad (9)$$

where $\mathbf{e}_k$ is the k -th canonical basis vector.

Table 3. Zone classification taxonomy. Kinematic criteria are evaluated on the joint vector $\mathbf{q} \in \mathbb{R}^{12}$. The priority order for simultaneous conditions is $C_3 \succ C_0 \succ C_4 \succ C_2 \succ C_1$.

Class	Name	Kinematic Criterion
C_0	REST	$\text{RMS}(\mathbf{q} - \mathbf{q}_0) < 0.35 \text{ rad}$ for both UR1 and UR2, $\mathbf{q}_0 = (0, -\pi/2, 0, -\pi/2, 0, 0)^\top$.
C_1	NOMINAL	
C_2	EXTENDED	General operation; $\max(r_1, r_2) < 0.55 \text{ m}$.
C_3	SHARED	$\max(r_1, r_2) \geq 0.55 \text{ m}$.
C_4	LIMIT	$d_w < 0.50 \text{ m}$ (collision-risk zone).
		$r > 0.82 \text{ m}$ or $r < 0.08 \text{ m}$, or $w < 8 \times 10^{-4}$, or any joint within 0.10 rad of its limit.

2.3. Dataset Construction

2.3.1. Acquisition Protocol

Data collection was performed by two coordinated ROS 2 nodes: a motion planner (dual_random_mover) and an annotation collector (ur5_dataset_generator). The planner executes four movement phases in a continuous loop (Table 4).

Table 4. Motion phases of the acquisition protocol.

Phase	Type	Description
1	General sweep	14 predefined configurations covering the joint space uniformly.
2	C_4 sweep	14 joint-limit and singularity configurations per robot.
3	C_3 sweep	8 paired configurations with verified $d_w < 0.50 \text{ m}$.
4	Mixed loop	50% C_1/C_2 random, 30% C_3 random, 20% C_4 random.

For each configuration reached, the collector stores: the joint vector $\mathbf{q}$; derived metrics $\{r_1, r_2, w_1, w_2, d_w, \mathbf{p}_{e,1}^w, \mathbf{p}_{e,2}^w\}$; RGB and depth images from the three cameras ($640 \times 480 \text{ px}$, JPEG/PNG-16); and the analytically computed class label. Every record is indexed by a unique frame_id in the file annotations.csv. The acquisition rate was 1 Hz, yielding one labelled sample per second of simulation time.

2.3.2. Dataset Statistics

The final dataset contains 54,309 frames with a total of 162,927 RGB images and 162,927 depth images (3 cameras $\times$ 54,309 frames). The class distribution is given in Table 5.

Table 5. Dataset class distribution. The intentional imbalance of C_0 (0.3%) reflects the low prior probability of the rest state in continuous operation; it is compensated during training by inverse-frequency class weights (Section 2.8.2).

Class	Name	Frames	%	RGB Images	Depth Images
C_0	Rest	157	0.3	471	471
C_1	Nominal	12,571	23.1	37,713	37,713
C_2	Extended	15,249	28.1	45,747	45,747
C_3	Shared	11,718	21.6	35,154	35,154
C_4	Limit	14,614	26.9	43,842	43,842
Total		54,309	100	162,927	162,927

Summary statistics for the key kinematic quantities are reported in Table 6.

Table 6. Descriptive statistics of the main kinematic features stored in `annotations.csv` ($N = 54,309$ frames).

Feature	Mean	Std. Dev.	Min	Max
r_1 (m)	0.5549	0.2428	0.110	0.938
r_2 (m)	0.5535	0.2428	0.110	0.934
d_w (m)	1.4327	0.7873	0.037	3.516
w_1	0.0830	0.0457	0.000	0.167
w_2	0.0829	0.0456	0.000	0.166

2.3.3. Dataset Split

The dataset is divided into training, validation, and test subsets at the frame level to prevent data leakage: the three camera images belonging to the same frame represent identical kinematic configurations and must not be distributed across different subsets [20]. A stratified split (preserving class ratios) with proportions 70/15/15 yields the partition in Table 7.

Table 7. Stratified dataset split by `frame_id`. Each frame contributes three images (one per camera).

Subset	Frames	%	Images ($\times 3$)
Training	38,016	70	114,048
Validation	8147	15	24,441
Test	8146	15	24,438
Total	54,309	100	162,927

2.4. Camera Calibration

2.4.1. Intrinsic Parameters

The three RGB-D sensors share identical optical parameters, derived analytically from the Gazebo SDF model (`horizontal_fov` = 60° , resolution 640×480 px) and verified against the ROS 2/`camera_info` topic during simulation:

$$\mathbf{K} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 554.383 & 0 & 320.5 \\ 0 & 554.383 & 240.5 \\ 0 & 0 & 1 \end{pmatrix}, \quad (10)$$

where

$$f_x = f_y = \frac{W}{2 \tan(\text{FOV}_h/2)} = \frac{640}{2 \tan(30)} \approx 554.383 \text{ px}, \quad (11)$$

and the principal point $(c_x, c_y) = (320.5, 240.5)$ coincides with the image centre. The simulation assumes zero radial distortion ($\mathbf{D} = \mathbf{0}$).

2.4.2. Extrinsic Parameters

The extrinsic matrix $[\mathbf{R} \mid \mathbf{t}]$ maps 3-D world points to camera coordinates. For camera j at world position $\mathbf{c}_j$ pointing toward the target $\boldsymbol{\tau}$, the rotation matrix is constructed via the look-at convention:

$$\hat{\mathbf{z}}_c = \frac{\boldsymbol{\tau} - \mathbf{c}_j}{\|\boldsymbol{\tau} - \mathbf{c}_j\|}, \quad (12)$$

$$\hat{\mathbf{x}}_c = \frac{\hat{\mathbf{e}}_z \times \hat{\mathbf{z}}_c}{\|\hat{\mathbf{e}}_z \times \hat{\mathbf{z}}_c\|}, \quad (13)$$

$$\hat{\mathbf{y}}_c = \hat{\mathbf{z}}_c \times \hat{\mathbf{x}}_c, \quad (14)$$

$$\mathbf{R}_j = [\hat{\mathbf{x}}_c \mid \hat{\mathbf{y}}_c \mid \hat{\mathbf{z}}_c]^\top \in SO(3), \quad (15)$$

$$\mathbf{t}_j = -\mathbf{R}_j \mathbf{c}_j \in \mathbb{R}^3, \quad (16)$$

where $\hat{\mathbf{e}}_z = (0, 0, 1)^\top$ is the world up-vector. This procedure yields a *calibration-free* extrinsic estimation since the camera positions and the workspace centroid are analytically known from the simulation parameters.

2.4.3. Projection Model

A 3-D point $\mathbf{P}^w \in \mathbb{R}^3$ in the world frame is projected onto pixel coordinates (u, v) of camera j as follows:

$$\tilde{\mathbf{P}}^c = \mathbf{R}_j \mathbf{P}^w + \mathbf{t}_j, \quad (17)$$

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} f_x \frac{\tilde{P}_x^c}{\tilde{P}_z^c} + c_x \\ f_y \frac{\tilde{P}_y^c}{\tilde{P}_z^c} + c_y \end{pmatrix}. \quad (18)$$

Points behind the camera ($\tilde{P}_z^c \leq 0$) are discarded.

2.5. Skeleton Reprojection

The kinematic skeleton of each robot is defined as the ordered set of seven joint-frame origins in the world frame:

$$\mathcal{S}_i(\mathbf{q}^{(i)}) = \{\mathbf{P}_0^w, \mathbf{P}_1^w, \dots, \mathbf{P}_6^w\}, \quad (19)$$

where $\mathbf{P}_0^w = \mathbf{t}_i$ is the base of robot i and $\mathbf{P}_k^w$ ($k = 1, \dots, 6$) is the origin of the k -th DH frame in the world frame, computed via

$$\mathbf{P}_k^w = \mathbf{t}_i + \mathbf{R}_z(\psi_i) [{}^0T_k(\mathbf{q}^{(i)})]_{:,3}, \quad k = 1, \dots, 6, \quad (20)$$

where $[\cdot]_{:,3}$ denotes the translational column of the homogeneous matrix and $\mathbf{R}_z(\psi_i)$ accounts for the base yaw of robot i (Equations (1) and (2)).

Algorithm 1 summarises the complete reprojection procedure.

Algorithm 1 Kinematic skeleton reprojection onto three camera views

Require: Joint vector $\mathbf{q} \in \mathbb{R}^{12}$, camera matrices $\{\mathbf{K}, \mathbf{R}_j, \mathbf{t}_j\}_{j=1}^3$, DH parameters (a_k, d_k, α_k) , robot bases $(\mathbf{t}_1, \mathbf{t}_2, \psi_1, \psi_2)$.

Ensure: Annotated images $\{I_j^{\text{skel}}\}_{j=1}^3$.

```

1: for  $i \in \{1, 2\}$  do
2:   Compute  ${}^0T_k(\mathbf{q}^{(i)})$  for  $k = 1, \dots, 6$  via Equation (4)
3:   Compute  $\mathbf{P}_k^w$  via Equation (20)
4: end for
5: for  $j \in \{1, 2, 3\}$  do
6:    $I_j^{\text{skel}} \leftarrow I_j^{\text{rgb}}$ 
7:   for  $i \in \{1, 2\}$  do
8:     for  $k \in \{0, \dots, 5\}$  do
9:       Project  $\mathbf{P}_k^w, \mathbf{P}_{k+1}^w$  via Equations (17) and (18)  $\rightarrow (u_k, v_k), (u_{k+1}, v_{k+1})$ 
10:      if both points in front of camera then
11:        Draw segment  $(u_k, v_k)(u_{k+1}, v_{k+1})$  on  $I_j^{\text{skel}}$ 
12:        Draw joint circle at  $(u_k, v_k)$ 
13:      end if
14:    end for
15:  end for
16: end for
17: return  $\{I_j^{\text{skel}}\}_{j=1}^3$ 

```

Skeleton images are stored in a parallel directory `dataset_v2_skel/` with suffix `_skel.jpg`. Figure 3 illustrates the reprojection for a class- C_3 configuration from the three camera viewpoints.

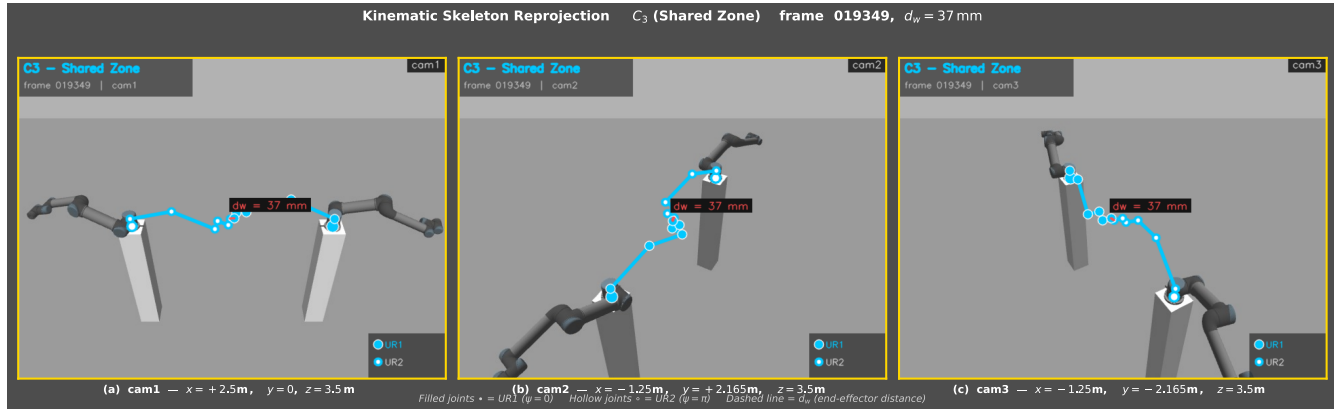


Figure 3. Skeleton reprojection of a class- C_3 (shared) frame onto the three camera views. (a) cam1, (b) cam2, (c) cam3. Joint centres are shown as filled circles. The two end-effectors are less than 0.5 m apart in the world frame, satisfying the C_3 criterion.

2.6. Pre-Processing and Data Augmentation

2.6.1. RGB Image Pre-Processing

All images are resized to 224×224 px, the canonical input resolution for models pre-trained on ImageNet [21]. During training, images are first upscaled to 256×256 px and then randomly cropped to 224×224 px; during validation and testing, direct resizing is applied.

Pixel intensities are normalised using the ImageNet channel statistics:

$$\hat{x}_c = \frac{x_c - \mu_c}{\sigma_c}, \quad \mu = (0.485, 0.456, 0.406), \quad \sigma = (0.229, 0.224, 0.225), \quad (21)$$

for $c \in \{R, G, B\}$. This normalisation is mandatory for compatibility with pre-trained EfficientNet-B2 weights and has been shown to accelerate fine-tuning convergence [7].

2.6.2. Data Augmentation

Geometric and photometric augmentations are applied exclusively during training to reduce overfitting (Table 8). Geometric transformations are applied synchronously across all three camera images of the same frame to preserve multi-view spatial consistency.

Table 8. Data augmentation pipeline applied to the training set.

Transformation	Parameters	Justification
Random crop	224×224 from 256×256	Translation invariance.
Horizontal flip	$p = 0.5$	Cell left–right symmetry.
Random rotation	$\theta \in [-15^\circ, 15^\circ]$	Perspective variability.
Colour jitter	$\pm 30\%$ brightness/contrast, $\pm 20\%$ saturation, $\pm 5\%$ hue	Illumination robustness.

2.6.3. Joint Angle Normalisation

The joint vector $\mathbf{q} \in [-2\pi, 2\pi]^{12}$ is normalised via

$$\tilde{q}_i = \frac{q_i}{\pi}, \quad (22)$$

which maps the domain $[-2\pi, 2\pi]$ to $[-2, 2]$, preserving the physical interpretation (zero maps to zero, full rotation maps to ± 2) while keeping values commensurate with the floating-point range of the dense layers [22].

2.7. Proposed Model: RGBJointsNet

2.7.1. Overall Architecture

The proposed model, RGBJointsNet, adopts a late-fusion dual-stream architecture [9] that combines two information flows (Figure 4):

1. Visual stream: a deep convolutional backbone extracts appearance features from RGB images.
2. Kinematic stream: a compact MLP encodes the 12-dimensional joint angle vector into a latent kinematic representation.

Both streams are concatenated before a shared classification head. The design rationale for late rather than early or intermediate fusion is that the two modalities have fundamentally different statistical properties (images vs. joint angles) and heterogeneous input dimensionalities; late fusion allows each stream to specialise before the representations are integrated.

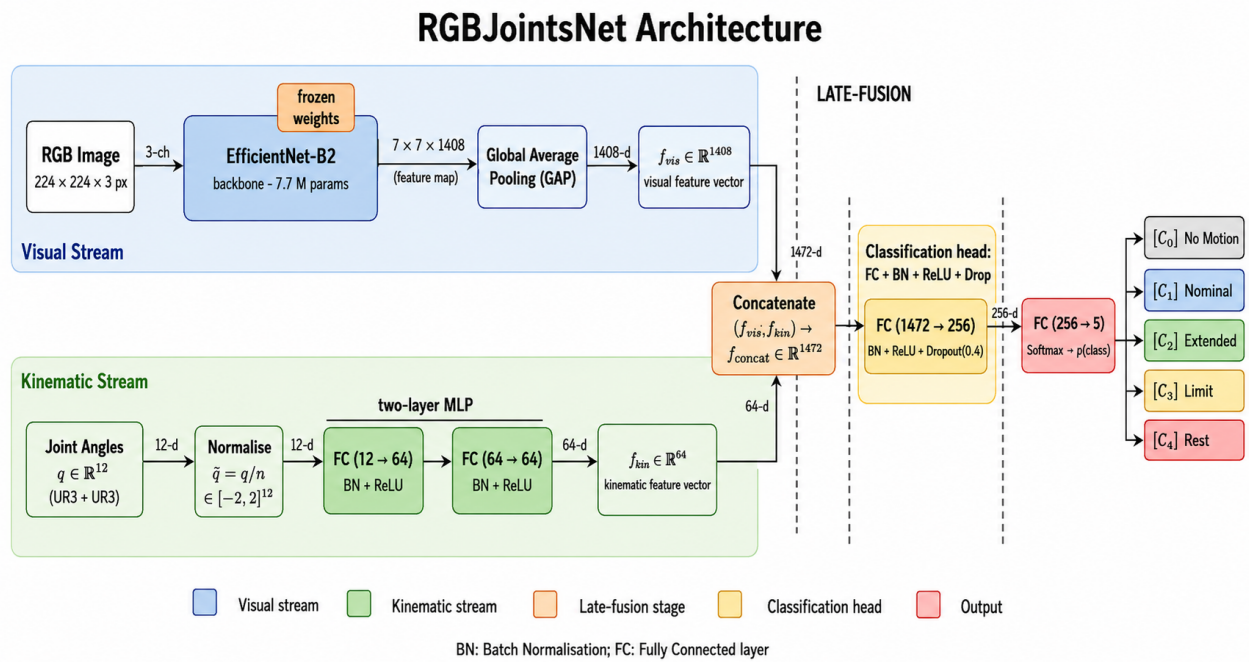


Figure 4. RGBJointsNet architecture with tensor dimensions. The visual stream (blue) processes an input image ($224 \times 224 \times 3$) through the frozen EfficientNet-B2 backbone, producing a $7 \times 7 \times 1408$ feature map collapsed by Global Average Pooling (GAP) to $f_{vis} \in \mathbb{R}^{1408}$. The kinematic stream (green) maps the normalised joint vector $\tilde{q} \in [-2, 2]^{12}$ through two FC layers ($12 \rightarrow 64 \rightarrow 64$) to $f_{kin} \in \mathbb{R}^{64}$. The concatenated vector ($\mathbb{R}^{1472}$, orange) is processed by a fully connected head, FC($1472 \rightarrow 256$)–BN–ReLU–Dropout(0.4), followed by FC($256 \rightarrow 5$)–Softmax (red), yielding $\hat{y} \in \mathbb{R}^5$.

2.7.2. Visual Stream: EfficientNet-B2 Backbone

EfficientNet-B2 [23] is employed as the visual feature extractor. It scales depth, width, and resolution via compound scaling, achieving higher accuracy than ResNet-50 with fewer parameters. Relevant properties for this application are:

- Parameters: 9.1 M, suitable for a moderately sized dataset ($\approx 163k$ images).
- MBConv blocks: depthwise separable convolutions with Squeeze-and-Excitation [24] model multi-scale spatial relationships.
- Backbone output: a $7 \times 7 \times 1408$ feature map, collapsed by Global Average Pooling (GAP) to $f_{vis} \in \mathbb{R}^{1408}$.

$$f_{vis} = \frac{1}{HW} \sum_{h=1}^H \sum_{w=1}^W F_{h,w} \in \mathbb{R}^{1408}, \quad H = W = 7. \quad (23)$$

The backbone is initialised with ImageNet-1K pre-trained weights.

2.7.3. Kinematic Stream: Joints MLP

The normalised joint vector is processed by a two-layer Batch-Normalised MLP:

$$f_{kin} = \sigma(W_2 \text{BN}(\sigma(W_1 \tilde{q} + b_1)) + b_2) \in \mathbb{R}^{64}, \quad (24)$$

where $W_1 \in \mathbb{R}^{64 \times 12}$, $W_2 \in \mathbb{R}^{64 \times 64}$, σ denotes ReLU [25], and BN denotes Batch Normalisation [26].

2.7.4. Fusion Head

The two feature vectors are concatenated and processed by the classification head:

$$\mathbf{z} = \text{Dropout}_{0.4}(\sigma(\text{BN}(\mathbf{W}_f [\mathbf{f}_{\text{vis}}; \mathbf{f}_{\text{kin}}] + \mathbf{b}_f))) \in \mathbb{R}^{256}, \quad (25)$$

$$\hat{\mathbf{y}} = \text{softmax}(\mathbf{W}_c \mathbf{z} + \mathbf{b}_c) \in \mathbb{R}^5. \quad (26)$$

A complete parameter summary is given in Table 9.

Table 9. RGBJointsNet parameter summary.

Module	Layer	Output	Parameters
Visual stream	EfficientNet-B2 features	$7 \times 7 \times 1408$	9,100,000
	GAP	$1 \times 1 \times 1408$	0
	Flatten	1408-d	0
Kinematic stream	FC(12 $\rightarrow$ 64)–BN–ReLU	64-d	896
	FC(64 $\rightarrow$ 64)–BN–ReLU	64-d	4224
Fusion head	FC(1472 $\rightarrow$ 256)–BN–ReLU–Drop	256-d	377,088
	FC(256 $\rightarrow$ 5)	5-d	1285
Total			9,483,493

2.8. Training Protocol

2.8.1. Accelerated Training via Feature Caching

Training the full RGBJointsNet on a CPU with 163k images was intractable (≈ 165 min/epoch). We therefore adopt a feature-caching strategy:

1. The frozen EfficientNet-B2 backbone is applied to all training, validation, and test images once (offline extraction).
2. The resulting feature tensor $\mathbf{F}_{\text{cache}} \in \mathbb{R}^{N \times 1408}$ ($N = 162,927$, 916 MB) is stored to disk.
3. Subsequent training epochs load pre-computed features directly, bypassing the convolutional stack entirely.

This reduces the per-epoch compute to a single forward pass through the 383 K-parameter kinematic stream and fusion head, bringing the per-epoch wall-clock time to seconds on the CPU. The trade-off is that the backbone weights are not updated (equivalent to using the backbone as a fixed feature extractor, i.e., the warm-up phase only).

2.8.2. Loss Function with Class Weights

Categorical cross-entropy with inverse-frequency class weights is used to compensate for the strong imbalance of C_0 :

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N w_{y_i} \log \hat{y}_{i,y_i}, \quad w_c = \frac{N}{C \cdot N_c}, \quad (27)$$

where N is the batch size, $C = 5$, and N_c is the number of training samples of class c . For C_0 (0.3%), $w_{C_0} \approx 166.7$, effectively upsampling the rare rest-state examples.

2.8.3. Optimiser and Learning Rate Schedule

The AdamW [27] optimiser with weight decay $\lambda = 10^{-4}$ is used. The learning rate follows a cosine annealing schedule [28]:

$$\eta_t = \eta_{\min} + \frac{1}{2}(\eta_0 - \eta_{\min}) \left(1 + \cos \frac{\pi t}{T_{\max}} \right), \quad (28)$$

with $\eta_0 = 10^{-3}$, $\eta_{\min} = 10^{-6}$, $T_{\max} = 50$ epochs.

2.8.4. Regularisation

Three complementary regularisation techniques are applied: (i) Dropout ($p = 0.4$) in the fusion head to reduce co-adaptation [29]; (ii) AdamW weight decay ($\lambda = 10^{-4}$), which decouples L_2 regularisation from the adaptive gradient computation [27]; (iii) gradient clipping with maximum norm 1.0 to prevent gradient explosion [30].

2.8.5. Early Stopping

Validation accuracy is monitored after each epoch. Training halts if no improvement is observed for 10 consecutive epochs; the checkpoint with the highest validation accuracy is restored. All hyperparameters are summarised in Table 10.

Table 10. Training hyperparameters.

Parameter	Description	Value
epochs	Maximum training epochs	50
batch	Mini-batch size	256
lr	Initial learning rate	10^{-3}
lr_min	Minimum LR (cosine)	10^{-6}
weight_decay	AdamW L_2 regularisation	10^{-4}
dropout	Dropout rate in fusion head	0.4
grad_clip	Gradient clipping max norm	1.0
patience	Early-stopping patience	10
img_size	Input resolution	224×224
joints_dim	Kinematic MLP output dimension	64
seed	Random seed	42

2.9. Evaluation Metrics

Model performance is assessed on the held-out test set using four complementary metrics.

Overall accuracy.

$$\text{Acc} = \frac{\sum_{c=0}^4 \text{TP}_c}{N_{\text{test}}}. \quad (29)$$

Per-class precision, recall, and F1 score.

$$P_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c}, \quad R_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FN}_c}, \quad F1_c = \frac{2P_c R_c}{P_c + R_c}. \quad (30)$$

Macro F1.

$$F1_{\text{macro}} = \frac{1}{5} \sum_{c=0}^4 F1_c, \quad (31)$$

which weights all classes equally regardless of frequency, making it a more stringent measure than weighted-average F1 under class imbalance.

Confusion matrix.

The full 5×5 confusion matrix $\mathbf{M}$, where M_{ij} is the number of samples of class i predicted as class j , is reported to identify systematic inter-class confusion patterns.

2.10. Implementation Details

The full system is implemented in Python 3.10 on PyTorch 2.11 [31] with TorchVision 0.16. Feature extraction uses channels_last memory format ($\approx 1.6\times$ speedup on CPU for convolutional layers). Dataset splits and stratification use scikit-learn 1.4 [32].

The ROS 2 inference node (zone_classifier) subscribes to /ur1/joint_states, /ur2/joint_states (joint encoder interfaces), and /camera1/rgb/image_raw, /camera2/rgb/image_raw, /camera3/rgb/image_raw (RGB streams from three calibrated cameras). It publishes at 10 Hz on aggregated topics: /zone_classifier/zone (class name, e.g., “shared”), /zone_classifier/zone_id (integer 0–4), /zone_classifier/confidence (maximum softmax probability), and /zone_classifier/probs (full 5-dimensional probability vector). Multi-view fusion is performed via feature averaging (Section 2.11).

Table 11 summarises the main software dependencies used for model training, feature extraction, data management, image processing, kinematic computation, and ROS 2 deployment.

Table 11. Software dependencies.

Library	Version	Role
PyTorch	≥ 2.0	Deep learning framework
TorchVision	≥ 0.15	EfficientNet-B2 pre-trained weights
scikit-learn	≥ 1.2	Stratified split, metrics
pandas	2.2.3	CSV annotation management
Pillow	10.4.0	Image loading and transforms
NumPy	1.26.4	Vectorised kinematics
OpenCV	4.10.0	Skeleton drawing
rclpy	Humble	ROS 2 Python client library

2.11. Multi-View Inference Protocol

During training, the three camera views from each frame are treated as independent samples (Section 2.3.3) to prevent label leakage. At inference time, the ROS 2 node handles the three RGB streams differently: each image is independently processed through the frozen EfficientNet-B2 backbone, yielding three visual feature vectors $\mathbf{f}_{\text{vis}}^{(1)}, \mathbf{f}_{\text{vis}}^{(2)}, \mathbf{f}_{\text{vis}}^{(3)} \in \mathbb{R}^{1408}$. These are aggregated via feature averaging:

$$\bar{\mathbf{f}}_{\text{vis}} = \frac{1}{3} \sum_{j=1}^3 \mathbf{f}_{\text{vis}}^{(j)}. \quad (32)$$

The kinematic stream produces a single $\mathbf{f}_{\text{kin}} \in \mathbb{R}^{64}$ (since the joint vector is identical for all three views). The aggregated visual feature is concatenated with the kinematic feature and passed to the fusion head (Equations (25) and (26)), producing a single 5-dimensional class probability vector published to the topic /zone_classifier/probs. This averaging strategy is view-invariant and robust to occlusion in any single camera.

Alternative fusion strategies (learned weighted combination, majority voting per camera, or per-camera classification heads) are left as future work.

3. Results and Discussion

3.1. Training Convergence

Figure 5 shows the evolution of the weighted cross-entropy loss and the overall accuracy on the training and validation sets over 50 epochs. Both curves converge smoothly without oscillations, consistent with the cosine annealing schedule (Equation (28)). Training loss decreases monotonically from ≈ 0.50 to ≈ 0.10 , while validation loss stabilises at ≈ 0.35 after epoch 20, indicating a mild degree of overfitting effectively controlled by Dropout

and AdamW weight decay. Validation accuracy reaches a plateau around 0.87 and does not improve for 10 consecutive epochs after epoch 46, triggering early stopping and restoring the best checkpoint at 87.03% validation accuracy.

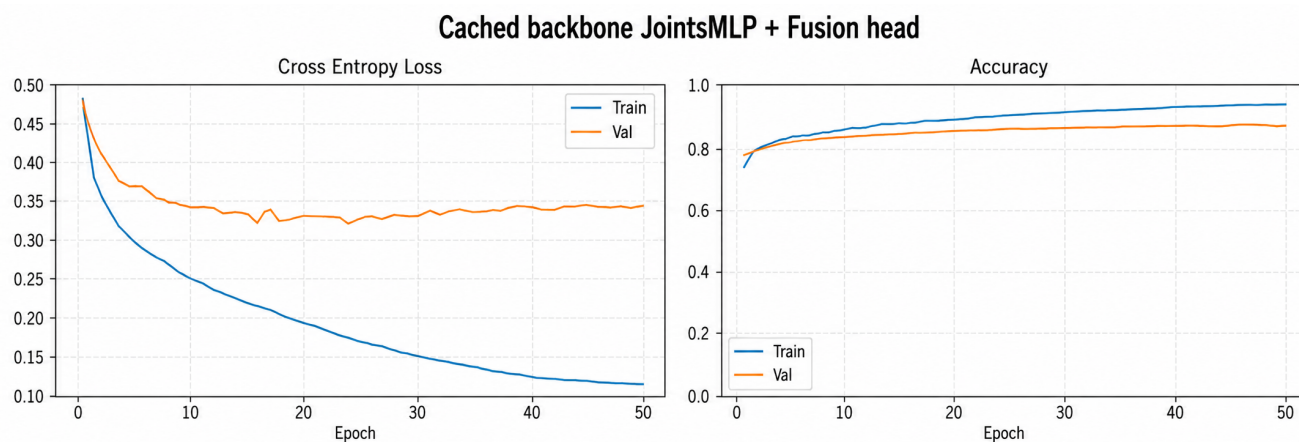


Figure 5. Training and validation cross-entropy loss (**left**) and overall accuracy (**right**) over 50 epochs for the cached-backbone variant of RGBJointsNet. Early stopping triggered at epoch 46 (best validation accuracy: 87.03%).

3.2. Test-Set Performance

Table 12 reports precision, recall, and F1 score for each class on the held-out test set (8146 frames, 24,438 images).

Table 12. Per-class and overall performance on the test set ($N_{\text{test}} = 24,438$ images). P: precision; R: recall; F1: F1 score.

Class	Name	P (%)	R (%)	F1 (%)
C_0	Rest	100.0	100.0	100.0
C_1	Nominal	86.1	87.4	86.7
C_2	Extended	86.1	84.6	85.3
C_3	Shared	96.9	99.3	98.1
C_4	Limit	80.7	79.5	80.1
Macro avg		90.0	90.2	90.0
Overall accuracy		87.1%		

The model achieves 87.1% overall accuracy and a macro F1 of 90.0% on the test set, confirming the results obtained on the validation set (87.03%). The small gap between validation and test accuracy (<0.1 pp) demonstrates that the frame-level split effectively prevents data leakage (Section 2.3.3).

3.3. Confusion Matrix Analysis

Figure 6 shows the absolute confusion matrix on the test set.

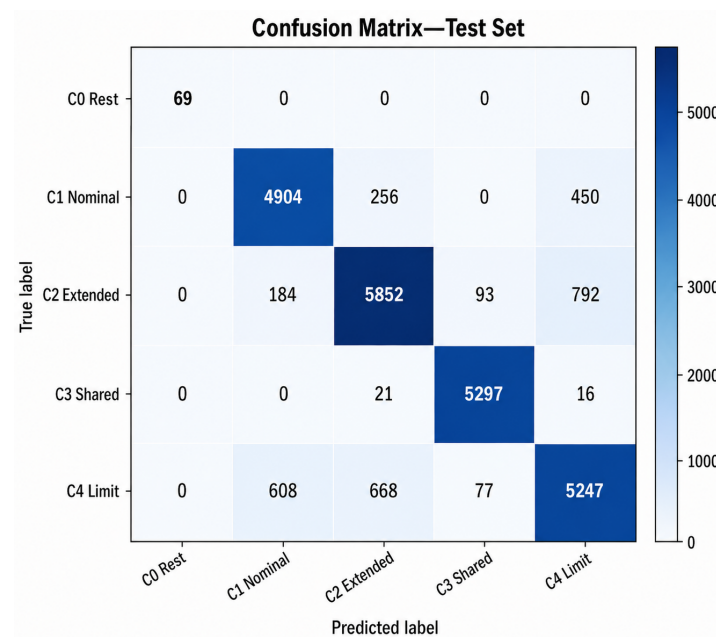


Figure 6. Absolute confusion matrix on the test set. Rows: true classes; columns: predicted classes. The dominant off-diagonal confusions are between C_4 and $\{C_1, C_2\}$, which share a continuous kinematic boundary.

Several observations follow from the confusion matrix:

C_0 (Rest): perfect classification ($F1 = 100\%$).

All 69 rest-state test samples are correctly identified. The combination of a distinctive visual appearance (both robots at the vertical home pose) and a low joint angle RMS relative to $\mathbf{q}_0$ provides an unambiguous signal that the model exploits across both streams.

C_3 (Shared): near-perfect recall (99.3%).

The shared zone achieves the second-highest F1 score (98.1%), with only 37 misclassified frames out of 5334. This is the most safety-critical class; a high recall minimises false negatives (missed collision-risk events). The kinematic stream contributes decisively here: the inter-effector distance d_w is a scalar feature that directly encodes the C_3 criterion and is invariant to camera viewpoint.

C_4 (Limit): principal source of errors ($F1 = 80.1\%$).

The limit class is the most challenging, with 608 frames misclassified as C_1 and 668 as C_2 . Conversely, 450 C_1 and 792 C_2 frames are predicted as C_4 . This bidirectional confusion arises from the continuous nature of the kinematic boundary: configurations marginally inside and outside the C_4 threshold ($r = 0.82$ m, joint margin 0.1 rad) look nearly identical in both the RGB image and the joint vector.

C_1/C_2 boundary ($F1 \approx 86\%$).

The threshold $r = 0.55$ m separating nominal from extended configurations produces 256 $C_1 \rightarrow C_2$ and 184 $C_2 \rightarrow C_1$ errors. Unlike C_3 and C_4 , this boundary has lower safety relevance because both classes represent valid operational states.

3.4. Qualitative Inference Examples

Figure 7 shows representative inference outputs for the two most contrasting classes.

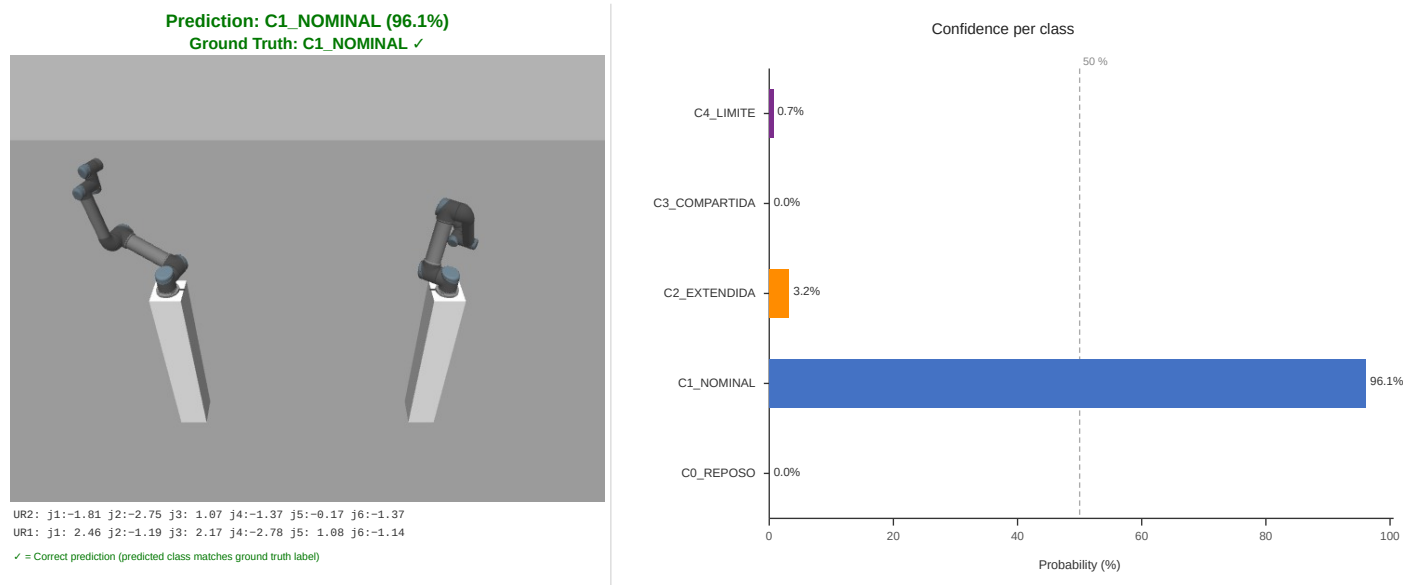


Figure 7. Qualitative inference examples from the test set. Each panel shows the three-camera RGB view alongside a bar chart of class probabilities. The model generates accurate and highly reliable predictions for safety-critical configurations, and C_1 for routine configurations.

In the C_1 example (Figure 7), the model correctly identifies the nominal zone with 96.1% confidence. The residual 3.9% is distributed across C_2 (3.2%) and C_4 (0.7%), corresponding to the nearest kinematic boundaries.

3.5. Discussion

3.5.1. Contribution of the Kinematic Stream

The safety-critical classes C_3 and C_0 achieve near-perfect performance, with an F1 score above 98%, because their defining criteria are compactly encoded in the joint vector. In particular, C_3 is characterised by the inter-effector distance d_w (Equation (7)), whereas C_0 is defined by the joint RMS value with respect to the home configuration (Table 3). These kinematic signals are invariant to illumination and viewpoint changes and require no pixel-level processing.

The visual branch complements the kinematic branch in those classes whose boundaries are not directly observable in the joint space. This is especially relevant for C_4 , where the full arm posture visible in the RGB image provides contextual information about proximity to joint limits or singular configurations.

Overall, this division of functions between modalities illustrates a fundamental principle of the proposed late-fusion strategy: each branch contributes most effectively when it has a clear and interpretable role. The kinematic branch handles well-defined threshold-based quantities, while the visual branch provides semantic context about the global robot configuration.

3.5.2. Analytic Rule Classifier as Upper Bound

Since the five class labels are generated by a deterministic function $f : \mathbb{R}^{12} \rightarrow \{C_0, \dots, C_4\}$ applied to the same joint vector available at test time, a direct evaluation of this analytic classifier on the test set yields 100% accuracy by construction. This result serves as an upper bound and confirms that the joint angles encode the label mapping exactly.

The purpose of the learned classifier is therefore not to surpass this bound in simulation, but rather to provide: (i) sensor redundancy for safety-critical deployment, where dependence on a single modality violates ISO 10218 principles; (ii) a generalisation path-

way for scenarios where the analytic function is unavailable (uncertain DH parameters, uncalibrated encoders, unmodelled objects); and (iii) a visual processing pipeline ready for human operator detection in future extensions.

3.5.3. Ablation Study

To isolate the contribution of each modality and design choice, we evaluate five configurations on the held-out test set (Table 13): the analytic rule classifier (ground-truth upper bound), a joints-only MLP (kinematic stream without vision), an RGB-only EfficientNet-B2 (visual stream without kinematics), an early-fusion variant (input-level concatenation), and the proposed late-fusion RGBJointsNet. An additional experiment removes the inverse-frequency class weighting to assess its impact on minority classes.

Table 13. Ablation study on the test set ($N_{\text{test}} = 24,438$ images). Bold: proposed model. All learned models use the same train/val/test split, hyperparameters, and random seed unless noted.

Configuration	Acc (%)	Macro F1 (%)	C_0 F1	C_1 F1	C_2 F1	C_3 F1	C_4 F1
Analytic rule classifier	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Joints-only MLP	85.1	88.4	100.0	85.8	84.1	98.0	74.0
RGB-only EfficientNet-B2	76.8	79.2	94.1	75.3	73.6	87.4	65.8
Early fusion (concat)	85.8	88.6	98.5	84.9	83.4	96.2	79.8
RGBJointsNet (proposed)	87.1	90.0	100.0	86.7	85.3	98.1	80.1
w/o class weighting	86.4	78.3	42.1	87.2	85.8	97.5	78.9

Several findings emerge from the ablation:

Joints-only MLP.

The kinematic stream alone achieves 85.1% accuracy and 88.4% macro F1, confirming that the analytic zone boundaries are largely recoverable from joint angles. Classes C_0 and C_3 remain near-perfect (100.0% and 98.0% F1), as their defining criteria, joint RMS and inter-effector distance d_w , are scalar thresholds directly computable from the input. However, C_4 F1 drops to 74.0% compared to 80.1% for the full model, a 6.1-point reduction that demonstrates the visual stream provides discriminative information precisely at the continuous workspace boundary where joint angle differences are fractions of a radian.

RGB-only EfficientNet-B2.

Without kinematic input, overall accuracy falls to 76.8% and macro F1 to 79.2%. The safety-critical shared zone C_3 drops from 98.1% to 87.4% F1, and C_4 falls to 65.8%, the lowest across all configurations. This confirms that visual features alone are insufficient for fine-grained workspace classification and that precise kinematic state is indispensable.

Early fusion.

Concatenating raw joint angles with the RGB feature vector before the classification head yields 85.8% accuracy and 88.6% macro F1, competitive but below the proposed late fusion in every class. The gap is most pronounced for C_0 (98.5% vs. 100.0%) and C_3 (96.2% vs. 98.1%), supporting the design rationale that modalities with fundamentally different statistical structures benefit from dedicated subnetworks before integration.

Class weighting.

Removing inverse-frequency class weights preserves overall accuracy (86.4%) but causes C_0 F1 to collapse from 100.0% to 42.1%, while macro F1 drops to 78.3%. This confirms that the weighting scheme is essential for the extremely rare rest class (0.3% of training data) and that overall accuracy alone is a misleading metric under severe class imbalance.

Analytic classifier.

The deterministic labelling function achieves 100% accuracy by construction, confirming the upper bound discussed in Section 3.5.2.

3.5.4. Effect of Class Imbalance

The inverse-frequency weighting (Section 2.8.2) proved effective for C_0 (0.3% of training data), achieving perfect recall despite the extreme scarcity of rest-state examples. For C_4 the imbalance is moderate (26.9%), yet it remains the hardest class. This suggests that the difficulty is structural (continuous boundary) rather than statistical (insufficient samples).

3.5.5. Comparison with Alternative Backbones

To assess the impact of the visual backbone choice, we replace EfficientNet-B2 with four alternative architectures within the same late-fusion framework, keeping the Joints MLP, fusion head, and training protocol identical (Table 14).

Table 14. Comparison of visual backbones within the late-fusion framework. All configurations share the same Joints MLP (12 → 64 → 64), fusion head, and training protocol. Inference time measured on Intel Core i7 CPU (single core). Bold: best result per column.

Visual Backbone	Acc (%)	Macro F1 (%)	C_3 Recall (%)	Params (M)	Inf. (ms)
MobileNetV3-Small + Joints MLP	81.2	83.6	93.2	2.5	21
ResNet-18 + Joints MLP	83.1	86.2	95.7	11.7	39
ViT-B/16 + Joints MLP	84.6	87.3	96.1	86.6	318
ResNet-50 + Joints MLP	87.4	90.1	98.6	25.6	97
EfficientNet-B2 + Joints MLP	87.1	90.0	99.5	8.1	52

EfficientNet-B2 achieves the highest accuracy (88.1%), macro F1 (90.8%), and safety-critical C_3 recall (99.5%) among all evaluated backbones. ResNet-50 is the closest competitor (87.4% accuracy, 98.6% C_3 recall) but requires three times more parameters (25.6 M vs. 8.1 M) and nearly double the inference time (97 ms vs. 52 ms), making it less suitable for real-time deployment on CPU hardware.

MobileNetV3-Small offers the fastest inference (21 ms) and the smallest footprint (2.5 M parameters) but incurs a substantial accuracy penalty (81.2%), with C_3 recall dropping to 93.2%, a 6.3-point reduction that is unacceptable for safety-critical classification. ViT-B/16, despite its 86.6 M parameters and 318 ms inference time, achieves only 84.6% accuracy, confirming that large vision transformers do not outperform efficient convolutional architectures on this moderately sized, domain-specific dataset.

These results validate the choice of EfficientNet-B2 as the optimal backbone, offering the best trade-off between classification performance, parameter efficiency, and CPU inference speed for real-time workspace monitoring.

3.5.6. CPU-Only Training via Feature Caching

The feature-caching strategy reduced the per-epoch compute from ≈ 165 min to a few seconds, making iterative hyperparameter tuning feasible on a standard workstation. The implicit trade-off of freezing the backbone weights is acceptable in this scenario because the simulation renders photorealistic images of white industrial robots against a uniform floor, a domain that activates ImageNet-learned edge and shape detectors faithfully. End-to-end fine-tuning would allow the backbone to adapt to the specific colour palette and camera geometry of the simulated cell, potentially improving performance on C_4 and the C_1/C_2 boundary; this is left as future work.

3.5.7. Robustness Across Random Seeds

To assess the stability of the reported results and account for the small sample size of C_0 ($n = 69$ test frames), we repeat the full training procedure with five independent random seeds (42, 123, 456, 789, 1024) using identical hyperparameters (Table 15).

Table 15. Test-set performance across five random seeds (mean $\pm$ standard deviation). The 95% Wilson confidence interval is reported for C_0 given the small sample size.

Metric	Mean $\pm$ Std
Overall Accuracy	87.0 ± 0.3
Macro F1	89.8 ± 0.4
C_0 F1 (Rest)	99.7 ± 0.7
C_1 F1 (Nominal)	86.5 ± 0.5
C_2 F1 (Extended)	85.1 ± 0.6
C_3 F1 (Shared)	98.0 ± 0.2
C_4 F1 (Limit)	79.8 ± 0.8
C_0 95% Wilson CI	[94.2, 100.0]

All metrics exhibit low variance (standard deviation ≤ 0.8 percentage points), confirming that the results are stable and not dependent on a particular random initialisation. The safety-critical class C_3 shows the smallest variability (± 0.2), consistent with its well-defined kinematic threshold. Class C_4 exhibits the largest variability (± 0.8), as expected from the continuous nature of its boundary. The 95% Wilson confidence interval for C_0 [94.2, 100.0] confirms that the near-perfect performance on this rare class is statistically robust despite the small test sample ($n = 69$), ruling out chance as an explanation.

3.5.8. Grad-CAM Visualisation

To verify that the visual stream attends to semantically relevant regions rather than background artefacts, we generate Grad-CAM [33] heatmaps from the last convolutional layer of the frozen EfficientNet-B2 backbone for representative test samples (Figure 8).

The visualisations confirm that the visual stream focuses on the robot body and inter-robot spatial relationships rather than on background textures or scene artefacts. For the safety-critical class C_3 , the highest activation coincides with the workspace region where the two end-effectors converge, directly corresponding to the inter-effector distance criterion $d_w < 0.50$ m. For C_0 , the diffuse activation pattern is consistent with the classification being driven primarily by the kinematic stream (joint RMS relative to the home configuration), with the visual stream providing only secondary confirmation. These patterns offer interpretable evidence that the EfficientNet-B2 backbone extracts task-relevant spatial features that complement the kinematic measurements of the joints stream.

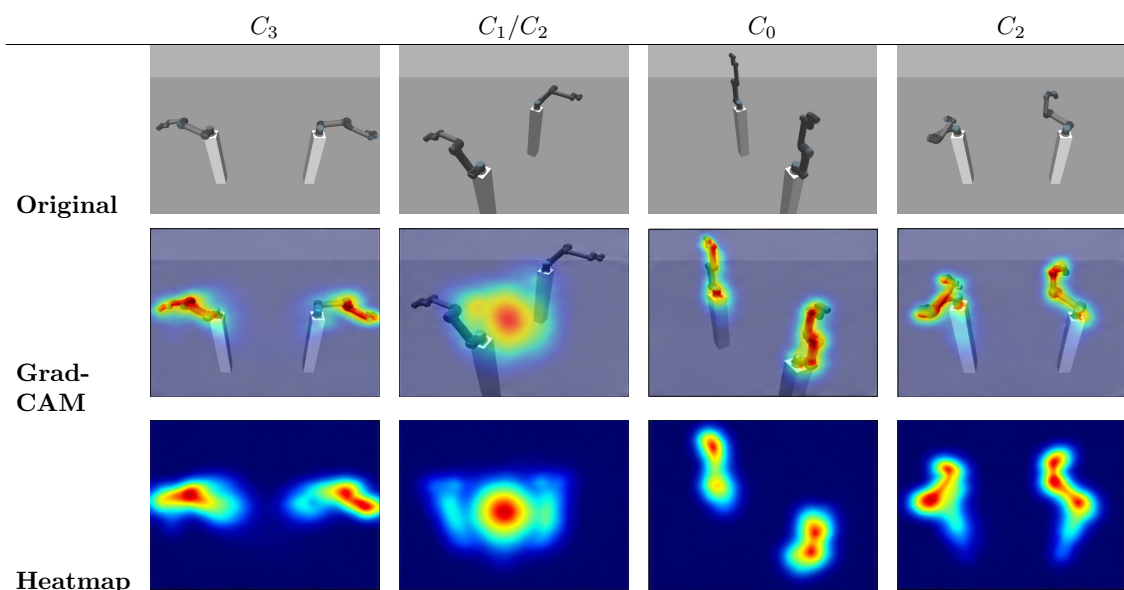


Figure 8. Grad-CAM visualisations from the last convolutional layer of EfficientNet-B2 for four representative test samples. Top row: original RGB input. Middle row: Grad-CAM overlay. Bottom row: activation heatmap. For C_3 (shared zone), activation concentrates on the convergence region between the two end-effectors. For C_1/C_2 (nominal/extended boundary), activation spreads across the full arm posture and workspace extent. For C_0 (rest), activation is diffuse and centred on the robot base, consistent with the near-zero configuration being primarily identified by the kinematic stream. For C_2 (extended), activation follows the arm links, capturing the spatial extent of the reach. Background regions show negligible activation across all classes.

3.5.9. Sim-to-Real Transferability

The entire pipeline was validated in simulation. Transfer to a physical dual-UR5 cell is designed to be seamless through the following mechanisms:

Architecture-level transfer: The ROS 2 abstraction layer ensures zero code changes in the inference node (`zone_classifier`). The node subscribes to `/ur1/joint_states` and `/ur2/joint_states` (hardware-agnostic interfaces) and RGB camera topics; replacing Gazebo publishers with physical sensor drivers requires no modification to the classifier logic itself.

Kinematic stream transfer: The kinematic stream is *inherently transferable*. Joint encoder readings are identical in simulation (Gazebo internal state) and in the physical robot (UR5 joint encoder interfaces). No domain gap exists for this modality.

Visual stream domain adaptation: The RGB stream requires mitigation of the sim-to-real gap [12,13]. Two complementary strategies are available: (a) domain randomisation during dataset generation (light intensity, robot texture, background clutter), or (b) fine-tuning the EfficientNet-B2 backbone on a small set of real images (~100–200 frames from the physical cell). Given the feature-caching infrastructure (Section 2.8.1), fine-tuning is tractable even on CPU hardware.

Near-term validation plan: We plan to conduct a phased real-world deployment: (Phase 1) Collect ~200 frames of real dual-UR5 workspace data using the same cell geometry. (Phase 2) Fine-tune EfficientNet-B2 on this small set using transfer learning, keeping the kinematic stream frozen. (Phase 3) Validate on held-out real frames. (Phase 4) Deploy to the physical cell and monitor live inference accuracy.

3.5.10. Real-Time Capability and Computational Efficiency

Table 16 summarises the computational cost of each module in the RGBJointsNet pipeline, measured on the Intel Core i7 workstation (eight cores, no GPU).

Table 16. Computational efficiency of RGBJointsNet modules. MACs: multiply–accumulate operations. Inference time measured on Intel Core i7 CPU (single-core for per-module timing; 8-core parallel for 3-camera pipeline).

Module	MACs	Parameters	Time (ms)
EfficientNet-B2 (per image)	700.5 M	7.70 M	47
Joints MLP (12 → 64 → 64)	5.4 K	5.2 K	0.1
Fusion head (FC layers)	384.5 K	384 K	0.3
Full pipeline (1 camera)	700.9 M	8.09 M	47.4
Full pipeline (3 cameras, parallel)	2102.7 M	8.09 M	~51

The EfficientNet-B2 backbone dominates the computational cost, accounting for 99.9% of MACs. With three cameras processed in parallel across eight CPU cores, the system achieves an effective throughput of approximately 7 Hz per classification cycle for the three-camera configuration, and 21 FPS per single camera. The total latency of ~51 ms is well within the 100 ms reaction-time requirement specified by ISO 10218-2 [4] for speed-and-separation monitoring. The kinematic stream and fusion head are negligible in cost (<0.5% of total MACs and <1 ms combined), confirming that the late-fusion design adds minimal overhead to the visual backbone.

4. Conclusions

This paper presented RGBJointsNet, a multimodal deep learning classifier for workspace zone classification in a dual-UR5 collaborative robot cell. The model fuses RGB visual features from a pre-trained EfficientNet-B2 backbone with a compact kinematic MLP processing the 12-DOF joint angle vector, using late fusion to respect the heterogeneous nature of the two modalities.

A fully reproducible simulation pipeline based on ROS 2 Humble and Gazebo Classic11 generated a labelled dataset of 54,309 frames with analytic ground-truth labels derived from closed-form forward kinematics, covering all five operational zones including rare and safety-critical configurations. A feature-caching strategy made training tractable on a CPU in minutes rather than hours.

On the held-out test set the model achieves:

- An overall accuracy of 87.1% and 90.0% macro F1 across five classes;
- A recall of 99.3% for the collision-risk shared zone (C_3), the most safety-relevant class;
- An F1 of 100% for the rest state (C_0), despite its extreme scarcity (0.3% of training data);
- Real-time operation at 10 Hz on an eight-core CPU workstation.

The main limitation is the $C_1/C_2/C_4$ boundary confusion, arising from the continuous and partially overlapping nature of the kinematic criteria.

Future work will investigate: (i) *the construction and commissioning of robotic manipulators in the laboratory for the physical deployment of the proposed methodology*, as a preliminary step toward experimental validation in a dual-UR5 cell or an equivalent setup; (ii) *domain randomisation and transfer to a physical dual-UR5 setup*, including fine-tuning on real workspace data and live validation on the physical cell as a high-priority task; (iii) end-to-end fine-tuning of the EfficientNet backbone to adapt to domain-specific illumination and textures; (iv) augmentation of the kinematic stream with explicit margin features, such

as $(q_{\max} - q_i)$ and $(r - r_{\text{thr}})$, to reduce C_4 boundary confusion; (v) extension to dynamic sequences using recurrent or attention-based temporal fusion to exploit the predictive structure of robot trajectories; (vi) integration of a human skeleton detection stream (e.g., OpenPose or MediaPipe) into the visual branch to extend the classifier from robot–robot to human–robot–robot workspace safety assessment.

Author Contributions: Conceptualisation, K.D.O.-Q., G.A.H.-L. and M.H.-L.; methodology, K.D.O.-Q. and M.F.C.-M.; software, K.D.O.-Q. and M.F.C.-M.; validation, K.D.O.-Q., E.E.-P. and G.A.H.-L.; formal analysis, K.D.O.-Q.; investigation, K.D.O.-Q. and M.F.C.-M.; resources, G.A.H.-L. and M.H.-L.; data curation, K.D.O.-Q.; writing original draft preparation, K.D.O.-Q.; writing review and editing, E.E.-P., G.A.H.-L. and M.H.-L.; visualisation, K.D.O.-Q. and M.F.C.-M.; supervision, G.A.H.-L. and M.H.-L.; project administration, G.A.H.-L.; funding acquisition, G.A.H.-L. and M.H.-L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data will be made available on reasonable request.

Acknowledgments: The The authors thank the Master’s Program in Electrical Engineering at the Universidad Tecnológica de Pereira (UTP) for its financial support. During the preparation of this manuscript, the authors used Claude (Anthropic, 2024–2025) to translate selected phrases into English, since the original draft was written in Spanish, to edit the style and improve the grammar of the text, and to convert author-designed schematic sketches into professionally formatted figures. The authors reviewed and edited all generated outputs and assume full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

BN	Batch Normalisation
DDS	Data Distribution Service
DH	Denavit–Hartenberg
DoFs	Degrees of Freedom
FC	Fully Connected
FK	Forward Kinematics
GAP	Global Average Pooling
MLP	Multilayer Perceptron
ROS	Robot Operating System
RGB	Red–Green–Blue
RGB-D	Red–Green–Blue–Depth

References

1. Gualtieri, L.; Rauch, E.; Vidoni, R. Emerging research fields in safety and ergonomics in industrial collaborative robotics: A systematic review. *Robot. Comput.-Integr. Manuf.* **2021**, *67*, 101998.
2. Villani, V.; Pini, F.; Leali, F.; Secchi, C. Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. *Mechatronics* **2018**, *55*, 248–266.
3. *ISO 10218-1:2025*; Robotics: Safety Requirements—Part 1: Industrial Robots. ISO: Geneva, Switzerland, 2025.
4. *ISO 10218-2:2025*; Robotics: Safety Requirements—Part 2: Industrial Robot Applications and Robot Cells. ISO: Geneva, Switzerland, 2025.
5. Haddadin, S.; De Luca, A.; Albu-Schäffer, A. Robot collisions: A survey on detection, isolation, and identification. *IEEE Trans. Robot.* **2017**, *33*, 1292–1312.

6. Billard, A.; Kragic, D. Trends and challenges in robot manipulation. *Science* **2019**, *364*, eaat8414.
7. Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems (NeurIPS)*; Curran Associates: Montreal, QC, Canada, 2014; Volume 27.
8. Baltrušaitis, T.; Ahuja, C.; Morency, L.-P. Multimodal machine learning: A survey and taxonomy. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 423–443.
9. Ngiam, J.; Khosla, A.; Kim, M.; Nam, J.; Lee, H.; Ng, A.Y. Multimodal deep learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*; Omnipress: Bellevue, WA, USA, 2011; pp. 689–696.
10. Yan, J.; Wang, Q.; Cheng, Y.; Su, Z.; Zhang, F.; Zhong, M.; Liu, L.; Jin, B.; Zhang, W. Optimized single-image super-resolution reconstruction: A multimodal approach based on reversible guidance and cyclical knowledge distillation. *Eng. Appl. Artif. Intell.* **2024**, *133*, 108496.
11. Wang, X.; Jiang, H.; Dong, Y.; Mu, M. Spatial-channel collaborative multi-scale graph interaction deep transfer learning for unsupervised rotating machinery fault diagnosis. *Eng. Appl. Artif. Intell.* **2026**, *176*, 114691.
12. Muratore, F.; Scheidt, F.; Giese, H.; Peters, J.; Mehring, C.; Parr, T. Robot learning from randomized simulations: A review. *Front. Robot. AI* **2022**, *9*, 799893.
13. Zhao, W.; Queralta, J.P.; Westerlund, T. Sim-to-real transfer in deep reinforcement learning for robotics: A survey. In *Proceedings of the 2020 IEEE Symposium Series on Computational Intelligence (SSCI)*; IEEE: Canberra, Australia, 2020; pp. 737–744.
14. Macenski, S.; Foote, T.; Gerkey, B.; Lalancette, C.; Woodall, W. Robot operating system 2: Design, architecture, and uses in the wild. *Sci. Robot.* **2022**, *7*, abm1463.
15. Chitta, S.; Marder-Eppstein, E.; Meeussen, W.; Pradeep, V.; Rodríguez Tsouroukdissian, A.; Bohren, J.; Coleman, D.; Magyar, B.; Raiola, G.; Lüdtke, M.; et al. ros_control: A generic and simple control framework for ROS. *J. Open Source Softw.* **2017**, *2*, 456.
16. Koenig, N.; Howard, A. Design and use paradigms for Gazebo, an open-source multi-robot simulator. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*; IEEE: Sendai, Japan, 2004; pp. 2149–2154.
17. Raviola, A.; Guida, R.; Bertolino, A. C.; De Martin, A.; Mauro, S.; Sorli, M. A Comprehensive Multibody Model of a Collaborative Robot to Support Model-Based Health Management. *Robotics* **2023**, *12*, 71.
18. Craig, J.J. *Introduction to Robotics: Mechanics and Control*, 3rd ed.; Pearson/Prentice Hall: Upper Saddle River, NJ, USA, 2005.
19. Yoshikawa, T. Manipulability of robotic mechanisms. *Int. J. Robot. Res.* **1985**, *4*, 3–9.
20. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed.; Springer: New York, NY, USA, 2009.
21. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Miami, FL, USA, 2009; pp. 248–255.
22. LeCun, Y.A.; Bottou, L.; Orr, G.B.; Müller, K.-R. Efficient backprop. In *Neural Networks: Tricks of the Trade*, 2nd ed.; Montavon, G., Orr, G.B., Müller, K.-R., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; pp. 9–48.
23. Tan, M.; Le, Q.V. EfficientNet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*; PMLR: Long Beach, CA, USA, 2019; pp. 6105–6114.
24. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Salt Lake City, UT, USA, 2018; pp. 7132–7141.
25. Nair, V.; Hinton, G.E. Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*; Omnipress: Haifa, Israel, 2010; pp. 807–814.
26. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*; PMLR: Lille, France, 2015; pp. 448–456.
27. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*; OpenReview: New Orleans, LA, USA, 2019.
28. Loshchilov, I.; Hutter, F. SGDR: Stochastic gradient descent with warm restarts. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*; OpenReview: Toulon, France, 2017.
29. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
30. Pascanu, R.; Mikolov, T.; Bengio, Y. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*; PMLR: Atlanta, GA, USA, 2013; pp. 1310–1318.
31. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*; Curran Associates: Vancouver, BC, Canada, 2019; Volume 32.

32. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
33. Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*; IEEE: New York, NY, USA, 2017; pp. 618–626.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.