Towards Vision-based Intersection Navigation: Explainable Insights from Synthetic and Real-World Model Adaptation

Yehan Kodithuwakku
Department of Computer Science and Engineering
University of Moratuwa
Moratuwa, Sri Lanka
kodithuwakkuhkayd.24@uom.lk

Chathuranga Hettiarachchi
Department of Computer Science and Engineering
University of Moratuwa
Moratuwa, Sri Lanka
hachathuranga@uom.lk

Sulochana Sooriyaarachchi
Department of Computer Science and Engineering
University of Moratuwa
Moratuwa, Sri Lanka
sulochanas@uom.lk

***Abstract*—Vision-based navigation offers an infrastructure-free substitute to GPS and external positioning systems, making it a worthy contender for robots operating in indoor environments such as libraries and supermarkets. This work proposes a novel ego-centric vision waypoint based navigation framework by formulating intersection understanding as a multi-label concept learning problem. To ensure a model can be trained with limited data and still preserve the conceptual understanding, each intersection is decomposed into its fundamental directional components: left, right, and forward. Further, we augment the limited real data with synthetic visual data collected from a safer digital twin environment which replicates the real-world scenes. We show that, (i) Model trained using digital twin data alone can yield an F1-score of 0.6 in real-world predictions, compared to 0.5 from a random classifier. (ii) Introducing limited real-world training data can bring the model up to near-perfect accuracy with faster convergence. (iii)The multi-label classification of junctions provides insightful visual explanations ensuring reliability and generalizability of the conceptual framework.**

***Index Terms*—ego-centric vision, explainability, concept learning.**

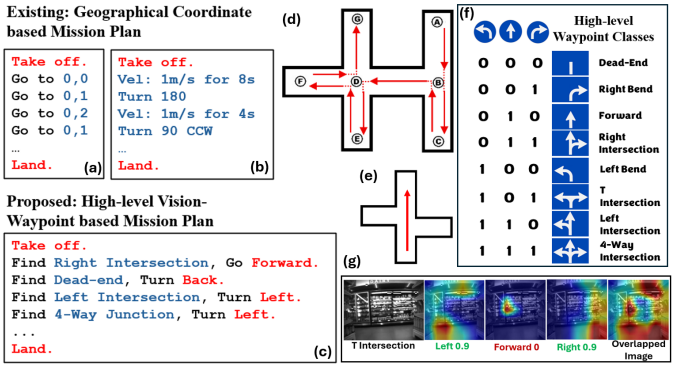


Fig. 1. Overall concept: (a) Global coordinate based mission plan (existing); (b) Global coordinate velocity setpoint based mission plan (existing); (c) Egocentric vision-waypoint based mission plan (proposed); (d) Mission plan of the drone; (e) Staggered junction—an example of a complex junction inferred zero-shot; (f) Orthogonal decomposition of junction vision-waypoints to left–forward–right navigability; (g) Validation of conceptual understanding of left–forward–right navigability of the proposed model with Grad-CAM.

I. Introduction

Autonomous navigation in robotics has witnessed a significant growth in recent years. Majority of this development is mainly used in self driving cars. Apart from autonomous ground vehicles, Unmanned Aerial Vehicles(UAVs) are becoming popular due to their wide range of applications. Among UAVs, research on nano-scale drones is an emerging topic due to variety of applications which are unattainable by standard size drones. Egocentric navigation from a first person perspective is the dominant strategy in the natural world, whereas the man-made systems tend to rely on external infrastructure such as Global Positioning System(GPS) satellites and fixed-position anchors, along with a global coordinate system. Humans naturally navigates in a real-world environment solely based on egocentric vision and their knowledge about the surrounding world (i.e. internal world model). Fascinatingly, a human can navigate in a first-person-view(FPV) drone by converting his visual perception to a few directional control an accomplish a high level task as illustrated in natural language in Fig. 1c. This is because all the required information is already available in the visual field. Egocentric vision represents the most intuitive and passive way of successfully navigating towards the goal only using the visual feedback. This navigation method does not require external anchors similar to Global Positioning Systems(GPS) because it is inherently independent of external infrastructure. Therefore, navigation can be accomplished without relying on a globally referenced coordinate or scale system by using this vision-centric technique. Vision offers an abundance of navigational clues that can be used to make decisions, such as identifying and interpreting intersections. We believe that a robot can also use these visual features to navigate in complex environments.

Recent research on world models like Video Joint Embedding Predictive Architecture(V-JEPA) [1] which is capable of high level understanding based planning and action execution would probably perform well in ego-centric navigation, but at

a higher computational cost. However, we believe that some of the high level key concepts such as turning left, turning right and going forward as illustrated in Fig. 1c can be derived to simplify a specific navigation task at a low computational cost.

Dronet project [2] is one prominent attempt on vision-based drone navigation, where they predict a collision probability and a yaw angle to avoid collision. It is an ego-centric, ResNet based Convolutional Deep Neural Network which takes a grayscale 200x200 frame [2] as the input. Later, The Parallel Ultra Low Power(PULP) research group carried out a number of studies with an emphasis on autonomous navigation for nano-scale drones. The Crazyflie platform is particularly well-liked because of its nano-scale [3], lightweight design, open-source control stack, and expandability with add-on decks. The AI-deck provides a 320x320 HM01B0 monochrome camera designed for low-power vision tasks. For navigation in nano-scale drones, PULP Dronet has been identified as the state of the art [4]. However, without a high level understanding about the decision points such as junctions, DroNet formulates the navigational problem as an obstacle avoidance task, rather than a waypoint based planned navigation task. The authors specifically pointed out that the model in the original DroNet research [2] chose a random route at intersections, which is not enough. PULP DroNet's steering output fails in intersection scenarios, as it cannot reason about multiple possible navigation directions simultaneously, highlighting a key navigable limitation in the state of the art nano-scale drones which motivates our work.

Navigating in complex environments such as intersections is a multi layered task where i. the high level vision-based waypoints need to be detected, and ii. the navigation decision is made. The decision can be either based on a pre-planned mission plan, or as a high-level real time piloting decision, where the autonomous robot awaits a decision from the pilot after reaching the decision point. An example mission planning and navigation scenario is given in Fig. 1d, where the mission plan of the drone starts from the coordinate A and ends at the coordinate G.

In contrast to the classical coordinate based mission plan depicted in Fig. 1a,b, we propose a visual waypoint based mission plan Fig. 1c. In traditional coordinate based navigation trajectory is defined by position or velocity points. In contrast, proposed conceptual framework uses high-level ego centric vision-based mission plan as described in Fig. 1c. The actions that the drone should take during an intersection is highlighted in red color.

In this paper, we focus on the first part of the problem, which is vision-based waypoint detection(perception-based). Among the limited previous work on intersection detection [5]–[7], the reported detection accuracies are promising. However, previous work cannot facilitate complex junctions such as staggered junctions shown in Figure 1e. Moreover, the conceptual understanding of the left, forward and right navigational affordances has not been considered, and the models lack explainability. We argue that the high-level waypoints should be the intersections, which can be further decomposed based on the concepts of left, forward and right navigability. This supports complex junction navigation. The Fig. 1f represents the orthogonal decomposition of the high-level intersection classes which can be used for mission planning, Fig. 1c. We have used Gradient-weighted Class Activation Mapping(Grad-CAM) heatmap [8] to verify the correct conceptual understanding of directions by the model, as depicted in Fig. 1g.

In our research, the key contributions are:

1) Training a vision-based model which conceptually understands high-level concepts of various intersections that can be used as navigational waypoints
2) Illustrating multi-label classification, achieved by decomposing high-level intersection categories outperforms conventional multi-class classification in intersection identification.
3) Verifying the actual conceptual learning using vision-based Artificial Intelligence(AI) explainable learning techniques.
4) Evaluating the generalization capability of the proposed model across unseen real-world data.
5) Collecting and publishing both synthetic and real-world dataset for intersection-level navigation tasks.

II. Related Work

The researchers approached the intersection identification task from several perspectives. Giusti et al. [9] classified drone photographs into three types (turn left, turn right, and go straight) so that a quadrotor could follow an outdoor hiking course. Authors have used a deep neural network with 10 layers with three output neurons as a classifier for the task.

Garcia et al. [6] proposed a method which use typical vision techniques with geometric shape analysis. First, video frames are captured from the drone's front camera. Next, significant lines in the scene are extracted using Hough Line Transform and Canny edge recognition. After that, these lines are processed to find right triangles created by the intersection of the floor and walls, which are used to accurately identify upcoming intersections. The size of these intersection triangles is analyzed, particularly the vertical side, which inversely correlates with the distance to the intersection. Advancing his previous research to address the limitation of the environmental lighting condition he proposed a Convolutional Neural Network(CNN) classifier which runs on the base station to classify intersections and dead-ends [10]. Garcia et al. [7] further have framed the intersection detection problem as an object detection task. They have labeled 1,484 photos from a wide range of corridors, with bounding boxes for different intersection types, doors, poster boards etc. A You Only Look Once(YOLO) based object detection model have been used to successfully detect the intersections.

Padhy et al. [11] also have reported successful likelihood estimations for class labels such as stop, shift left, shift right, and move forward, which can be used for autonomous maneuvering of the drone in corridor environments. The authors have used Dense-Net-161 and fed real-time images from a front facing camera as the input. Mansouri et al. [5] suggested an

approach that uses a monocular camera and CNNs, specifically utilizing transfer learning with AlexNet, to identify tunnel junctions in underground mines. The CNN is based on a four-class classification scheme of left, right, left & right and no junction. The authors followed a transfer learning approach since the real-world data was limited.

III. Methodology

The entire workflow of our suggested method is described in this section. As mentioned in Subsection III-A and III-B, we first select real-world environments and build digital twin supermarket environments. Subsection III-C then goes into detail into the data collection procedure used in both real-world and simulated settings. Lastly, Subsection III-D presents the experimental setups and model training pipeline. Our implementation, experimental framework, performance comparison, detailed results, and supplementary materials are available on GitHub[1].

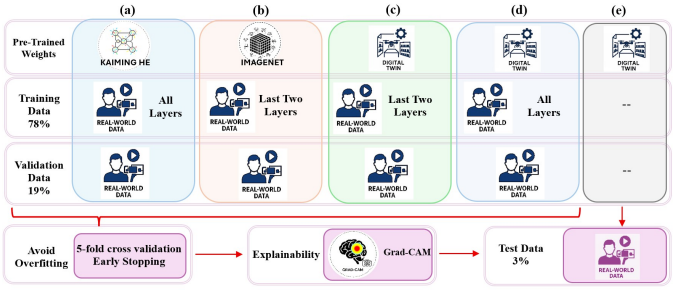


Fig. 2. Experimental design workflow of the proposed methodology. Five experiments were conducted with different weight initialization and training procedures. The dataset split used was 78% for training, 19% for validation, and 3% for testing. (a) Kaiming He initialization; (b) ImageNet initialization; (c) Digital twin initialization with training on the last two layers; (d) Digital twin initialization with training on all layers; (e) Digital twin initialization tested on real data with no real-world training.

A. Real-World Environmental Setup

We used the AI-deck's monochrome camera to manually gather grayscale data at 5 frames per second in two supermarkets and a library to make sure the model learnt real-world features and textures. During the data collection process, a custom tool was created to easily name images into its appropriate class. To provide wide visual variation, we recorded a variety of supermarket and library aisle settings, such as different lighting, intersections, walls, and bookshelf layouts. We determined the middle waypoint for every intersection and chose a data point that was 1.15 m before mid-waypoint. This distance guarantees that the drone can see the rack edges clearly inside its field of vision, allowing it to make the right left or right judgments without running the risk of colliding. Nevertheless, this method of gathering data is tedious, entirely manual, and may damage the drone while it is in use.

[1]https://github.com/yeko31/ego-intersect

B. Digital Twin Environmental Setup

The digital twin environment offers complete control over parameters, repeated experimentation, and secure, collision-free data collection advantages that are not achievable in real-world configurations. It is possible to accurately change the robot's altitude, intersection distance, camera height, and field of view. We focused on aisle structures that resemble supermarkets and libraries because of their neat 90° layouts and practical applicability. The AI-deck's grayscale camera and 87° field of view were matched to the virtual Crazyflie camera in Webots simulation platform, resulting in realistic pictures. To train our navigation model, we produced a variety of data pertaining to lighting, texturing, rack configurations, and shadows using these parameterized digital twins.

Webots controllers were designed to automate the egocentric data collection pipeline which is time and resource consuming if done by manually. Data points are defined as 1.15 m before the middle waypoint of any intersection. The drone follows a predefined set of waypoints which were hard-coded into folders corresponding to each intersection class inside the simulated environment. At each data point, the drone performs a slight lateral wiggle to capture different views of the racks. In some waypoints, the drone performs yaw rotations to secure the egocentric view of the camera. All together there are 22 waypoints covering 7 different intersection types including forward pathways and bends. We only selected the simulated environments which has aisle widths of 1 m and 1.3 m to preserve the views of the racks before reaching to an intersection point.

C. Data Collection

In both synthetic and real-world environments, the data was gathered at two altitude levels. Scenes were first classified with seven high-level classifications such as left bends, left intersections, right bends, right intersections, t intersections, four-way intersections and straight class before being broken down into a single three-bit directional format that indicated whether left, right, and forward pathways were available. Contrast and blur augmentations, as well as the creation of artificial bends to boost samples, were used to rectify the imbalance in left and right intersections found in real-world data. In the end, 3929 training images, 100 real test images, and 2560 synthetic images were collected for the studies as described in Table I. For the real-world dataset, images for each high-level class were collected and prepared in approximately equal quantities to maintain class balance.

D. Model Training

For intersection detection, we tested both MobileNetV2 and ResNet50. ResNet50 was chosen for its robust image-classification capabilities [12], while MobileNetV2 was picked for its lightweight architecture appropriate for nano-scale drones. We only present the ResNet50 results in this paper due to space constraints. We modified MobileNetV2 and ResNet50 to take into account the requirements of the task and the characteristics of the dataset. The final fully connected

TABLE I
DATASET CONFIGURATION USED IN MULTI-CLASS AND MULTI-LABEL EXPERIMENTS

Data Type	Number of Samples
Multi-class – Train (Real)	3929
Multi-label – Train (Synthetic)	2560
Multi-label – Test (Real)	100
Multi-label – Train (Real)	3929

TABLE II
K-FOLD PERFORMANCE COMPARISON ACROSS EXPERIMENTS A–D

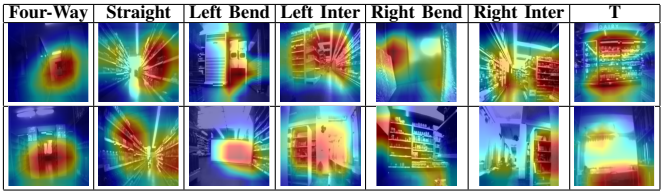
	Kaiming He (a)		ImageNet (b)		Digital Twin (Last 2) (c)		Digital Twin (All) (d)	
Fold	Best Val Acc	Epochs	Best Val Acc	Epochs	Best Val Acc	Epochs	Best Val Acc	Epochs
1	0.9898	10	1.0	6	0.9949	10	1.0	15
2	0.9707	7	1.0	4	0.9962	11	0.9987	16
3	0.9975	13	1.0	4	0.9822	8	0.9796	8
4	0.9835	11	1.0	6	0.9924	11	0.9707	8
5	0.9567	7	1.0	5	0.9924	12	0.9949	12

layers of the original models, which had been pre-trained on ImageNet, were swapped out for three classification heads, which matched the number of output classes in our dataset. We applied values of 0.2 color jitter for brightness and contrast as augmentation. All grayscale images were duplicated across three channels to ensure compatibility with the pretrained architectures, as both networks were initially intended for three-channel RGB inputs.

We first framed the task as a multi-class classification problem, assigning each image to one of seven intersection types: straight, left bend, left intersection, right bend, right intersection, four-way, and T-junction. We followed this method initially because most of the related work addressed this challenge as a multi class problem. We first trained all network parameters using the synthetic data collected from Webots environments in Section III-C. To train with real-world data, the weights and biases of the models were initialized using synthetic pretrained weights using the synthetic data. Training as a multi class classification included total data of 3929. Criterion was chosen as CrossEntropyLoss and Adam as an optimizer with a learning rate of 0.001. We froze all the layers except for the final two layers and continued training.

Multi class classification method has certain issues. For example, left intersection and left bend has shared concept which are captured as unrelated in this setting. Therefore, learning two different feature sets in this case, will not lead to the correct interpretation of the above intersection types because of the similarities. Another drawback is that the difficulty of categorizing a heterogeneous decision point into a single class as shown in Fig. 1e. Therefore, to solve the navigation problem, a multi label image classifier was developed which shows the possible pathways the drone can take. As summarized in the Fig. 1f, multi-hot binary label classification was used where each image could simultaneously include multiple directional indicators (left, forward, right). In this setting, we can still generalize to more complicated heterogeneous scenarios due to same concepts being shared. We believe that the use of multi-label targets can further extend to previously unseen and more complex environments by decomposing navigation decisions into orthogonal yet related concepts.

TABLE III
GRAD-CAM VISUALIZATION RESULTS FOR DIFFERENT INTERSECTION TYPES IN THE MULTI-CLASS CONFIGURATION (TWO SAMPLES FROM DIFFERENT LOCATIONS PER CLASS). PROPOSED IMPROVED VERSION IS GIVEN IN TABLE IV

Four-Way	Straight	Left Bend	Left Inter	Right Bend	Right Inter	T
						

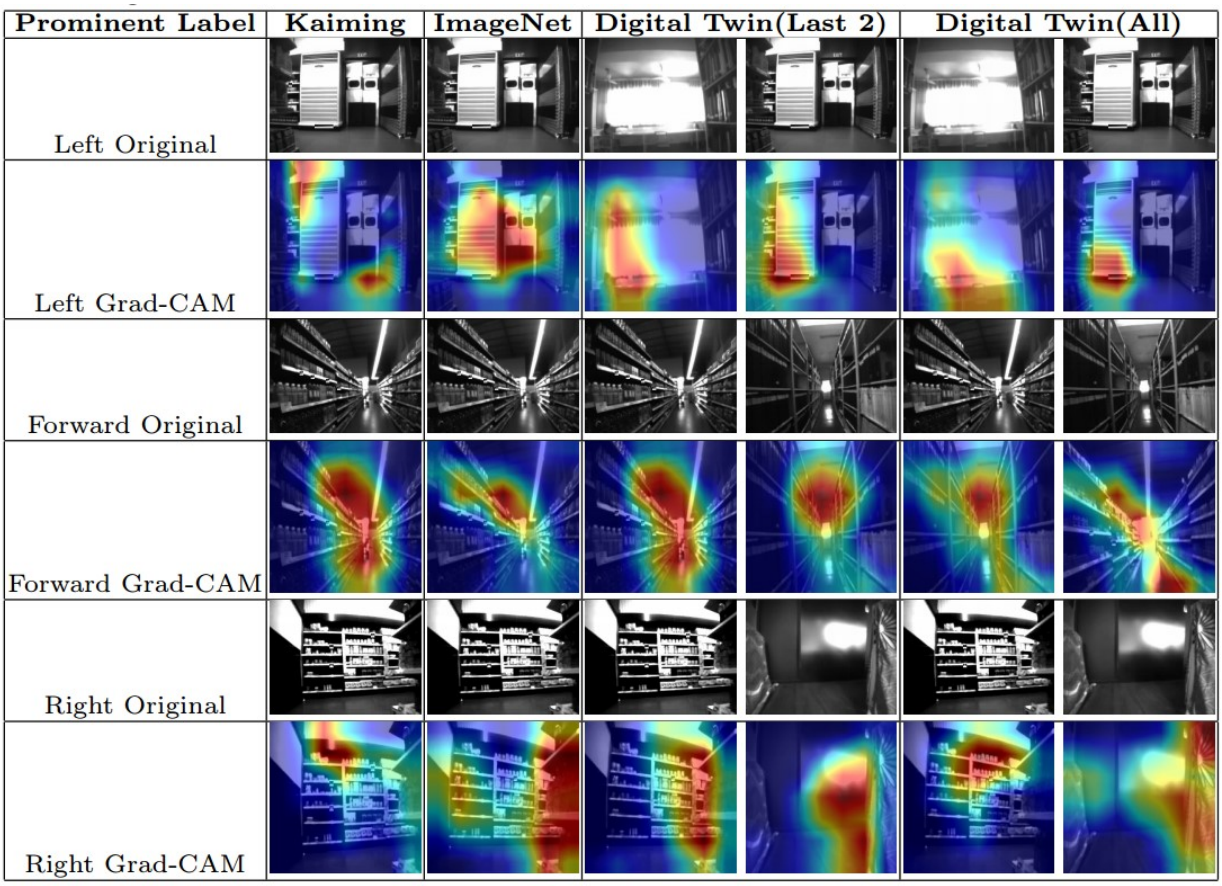
We have listed the main phases of our multi label class experiments in Fig. 2 that are done for both ResNet50 and MobileNetV2 architectures. Data distribution is explained under the Section III-C. The intention of using five fold cross validation is to observe the overfitting issue. As shown in Fig. 2, weight initialization was done in three main ways which are digital-twin, ImageNet and random. There are three stages in the experimental design framework. The first stage is the weight initialization phase. In digital twin approach, we train all layers from the scratch using the synthetic data collected from Webots simulation as describe in the Section III-B. The second stage is the real-world training phase. In this stage, some models were fully trained, others were partially trained, and a few were not trained at all under the real-world five-fold cross-validation setting. We set the criterion for BCEwithLogitsLoss and set the optimizer as Adam. Next, Grad-CAM was used to observe the significant areas of the input image which were used to predict the outcome. We used the model which achieved the highest validation accuracy across the folds for this. This allows the model to confirm that the model ignores background noise and instead pays attention to semantically significant areas like stock shelves, aisle borders, or intersection cues. Testing is the final stage where we use different metrics to observe the genralizability. All the configurations are available in the supplementary materials in Github.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

In Table III, we observe ResNet50 repeatedly misses salient visual cues to classify into different intersection types. For example, in T-junction scenarios, the network mostly concentrates on the top and bottom portions of the image instead of the structural junction features. Similar to this, attention is skewed toward right-edge features in left-bend circumstances, but the opposite behavior is seen in right-bend images. Further, in Table III, the areas highlighted are not uniform across different images belong to the same intersection. Hence, as discussed in Section III-D, we shifted the classification task from multi class to multi label.

In multi label configuration, we report the best validation accuracies in each fold and maximum number of epochs until early stopping in the Table II. Here, outcomes of Kaiming He initiation, training from scratch requires more epochs and exhibits slight instability in between folds. ImageNet weight

TABLE IV
GRAD-CAM VISUALIZATION RESULTS FOR THE MULTI-LABEL CONFIGURATION ACCORDING TO FIG. 2.



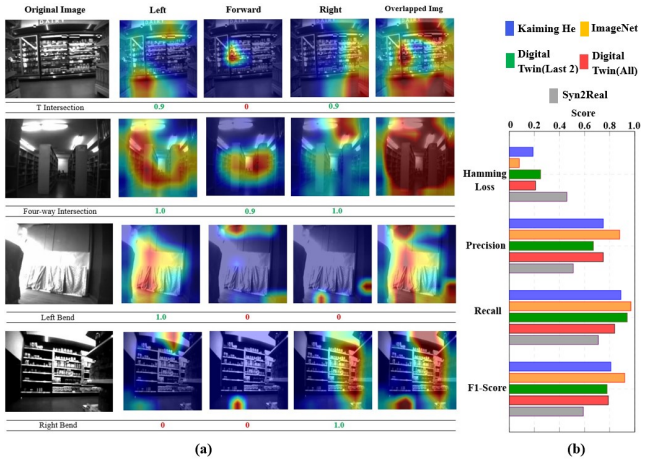


Fig. 3. (a) Successful left–forward–right mutually exclusive (orthogonal) conceptual understanding learned by the digital twin (2-layers) model, illustrated using Grad-CAM heatmaps across different junctions. Predicted probabilities are shown below each example. (b) Comparison of the five experiments across Hamming loss, precision, recall, and F1 score on the test data.

initialization demonstrated stronger and faster convergence compared to Kaiming He and Digital Twin initialization methods. ImageNet-initialized models regularly achieve perfect validation accuracy (1.0) in very few epochs across all folds. Digital twin last two configuration remains very high in most folds. A good trade-off between speed and accuracy is achieved by fine-tuning only the classifier (final layers), which maintains pretrained knowledge and adapts effectively.

Grad-CAM was utilized to confirm if the model recognized relevant concepts instead of random features. We observe that the Grad-CAM experimental configurations produce distinct activation patterns across configurations (a), (b), (c), and (d) in Fig. 2, as evident from the visualizations presented in Table IV. Grad-CAM heatmaps produced in relation to the most prominent label, that is, the class with the highest projected probability is presented in the tables. To demonstrate spatial consistency and variance, samples from diverse scenes are displayed for each label. Grad-CAM maps under configuration Kaiming He looks weaker feature localization. It's possible that early layers didn't develop strong spatial hierarchies, which made the heatmaps less interpretable. Compact and consistent activation zones are seen in configuration ImageNet, indicating more stable representations compared to configuration Kaimimg He. Explainability findings show that transfer learning increases both accuracy and visual interpretability. However, it is visible that in the Table IV, configuration ImageNet is unable to correctly identify the left visual cues.

Digital twin-based fine-tuning produces activation maps more closely aligned with the pertinent scene elements as illustrated in the final two columns of Table IV. The digital twin last two configuration performed better than the other configurations in all four experiments, resulting in Grad-CAM heatmaps that were sharper and more domain-focused. This configuration shows enhanced interpretability and task awareness since the attention regions corresponding to the left, right, and forward labels are well aligned with the key spatial locations.

We chose the top-performing model in explainability, the digital twin configuration trained on the last two layers, to look into this behavior in more detail. We examined several kinds of intersections and bends, as shown in Fig. 3a, and created heatmaps that matched their labels. The left, right, and forward heatmaps were then overlapped to generate composite visualizations. The predicted probabilities of the corresponding labels are shown by the color bar beneath each image. Interestingly, in both the T-junction and four-way intersection scenarios, the T-intersection mostly activates the left and right regions, whereas the four-way situation also displays activation (red regions) in the central area. In the same manner, the overlapped heatmaps for left and right bends only show the pertinent edge regions, illustrating the model's capacity to direct attention based on directional context.

Fig. 3b shows the comparison results on the test data. With the lowest Hamming Loss (0.08) and the highest F1 Score (0.92) of any configuration, the ImageNet weight initialization approach showed the best generalization performance. Our synthetic-to-real configuration shows that digital twin data can reasonably generalize to real-world images with a F1-score of 0.6 compared to a random classifier which is resulting 0.5. The performance can be further improved by incorporating real-world data, highlighting the importance of developing a digital twin when real datasets are limited.

V. Conclusion and Future Work

In this paper, we introduced an egocentric visual waypoint navigation that leverages vision-based intersection identification. Initially, we trained a model that can utilize high-level intersection types as navigation waypoints. Further, we reframed the task as multi-label classification by dividing high-level intersecting classes into directional labels, which enhances performance over multi-class approaches and more successfully captures shared features. Third, we used Grad-CAM to ensure that the network pays attention to semantically relevant features surrounding intersections and bends, ensuring that the model's conclusions are based on meaningful visual concepts rather than just numerical artifacts. The pretrained digital twin model which is trained on last two layers maintains consistently high accuracy across the training, validation, and testing datasets, while also exhibiting meaningful and interpretable feature localization when evaluated using Grad-CAM.

In future work, we plan to extend our work to the visual waypoint based navigation task. Further, we aim to develop an automated pipeline to transform real-world video into 3D simulation environments and improve the digital twin for safer, more effective validation. Finally, we plan to expand our study to challenging intersection types including Y-intersections, and improve the generalization across different spatial locations.

Acknowledgment

This research is funded by the University of Moratuwa Senate Research Committee Grant SRC/LT/2024/02. The authors acknowledge the support received from the LK Domain Registry in publishing this paper. We also thank Mr. Ignatious Peiris, Mr. Vihanga Wickramage, Mr. Sandun Dushyantha for their effort for data collection and synthetic environment building.

References

[1] M. Assran, A. Bardes, D. Fan, Q. Garrido, R. Howes, Mojtaba, Komeili, M. Muckley, A. Rizvi, C. Roberts, K. Sinha, A. Zholus, S. Arnaud, A. Gejji, A. Martin, F. R. Hogan, D. Dugas, P. Bojanowski, V. Khalidov, P. Labatut, F. Massa, M. Szafraniec, K. Krishnakumar, Y. Li, X. Ma, S. Chandar, F. Meier, Y. LeCun, M. Rabbat, and N. Ballas, "V-JEPA 2: Self-Supervised Video Models Enable Understanding, Prediction and Planning," 2025. [Online]. Available: http://arxiv.org/abs/2506.09985

[2] A. Loquercio, A. I. Maqueda, C. R. Del-Blanco, and D. Scaramuzza, "DroNet: Learning to Fly by Driving," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1088–1095, 2018.

[3] V. Niculescu, L. Lamberti, F. Conti, L. Benini, and D. Palossi, "Improving Autonomous Nano-Drones Performance via Automated End-to-End Optimization and Deployment of DNNs," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 11, no. 4, pp. 548–562, 2021.

[4] L. Lamberti, L. Bellone, L. Macan, E. Natalizio, F. Conti, D. Palossi, and L. Benini, "Distilling Tiny and Ultrafast Deep Neural Networks for Autonomous Navigation on Nano-UAVs," *IEEE Internet of Things Journal*, vol. 11, no. 20, pp. 33 269–33 281, 2024.

[5] S. S. Mansouri, P. Karvelis, C. Kanellakis, A. Koval, and G. Nikolakopoulos, "Visual Subterranean Junction Recognition for MAVs based on Convolutional Neural Networks," *IECON Proceedings (Industrial Electronics Conference)*, vol. 2019-Octob, pp. 192–197, 2019.

[6] A. Garcia, E. Mattison, and K. Ghose, "High-speed vision-based autonomous indoor navigation of a quadcopter," *2015 International Conference on Unmanned Aircraft Systems, ICUAS 2015*, pp. 338–347, 2015.

[7] A. Garcia, S. S. Mittal, E. Kiewra, and K. Ghose, "A Convolutional Neural Network Feature Detection Approach to Autonomous Quadrotor Indoor Navigation," *IEEE International Conference on Intelligent Robots and Systems*, no. July 2021, pp. 74–81, 2019.

[8] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization," *International Journal of Computer Vision*, vol. 128, no. 2, pp. 336–359, 2020.

[9] A. Giusti, J. Guzzi, D. C. Ciresan, F. L. He, J. P. Rodriguez, F. Fontana, M. Faessler, C. Forster, J. Schmidhuber, G. D. Caro, D. Scaramuzza, and L. M. Gambardella, "A Machine Learning Approach to Visual Perception of Forest Trails for Mobile Robots," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 661–667, 2016.

[10] A. Garcia, S. S. Mittal, E. Kiewra, and K. Ghose, "A convolutional neural network vision system approach to indoor autonomous quadrotor navigation," *2019 International Conference on Unmanned Aircraft Systems, ICUAS 2019*, no. May, pp. 1344–1352, 2019.

[11] R. P. Padhy, S. Verma, S. Ahmad, S. K. Choudhury, and P. K. Sa, "Deep Neural Network for Autonomous UAV Navigation in Indoor Corridor Environments," *Procedia Computer Science*, vol. 133, pp. 643–650, 2018. [Online]. Available: https://doi.org/10.1016/j.procs.2018.07.099

[12] J. Wan, B. Li, K. Wang, X. Teng, T. Wang, and B. Mao, "An Improved ResNet50 for Environment Image Classification," *Procedia Computer Science*, vol. 242, pp. 1000–1007, 2024. [Online]. Available: https://doi.org/10.1016/j.procs.2024.08.246