From Synthesis to Realism: Enhancing Logistics with Computer Vision and Domain-Adversarial Training

Quoc-Cuong Dang
Ha Noi University of Science and Technology
Hanoi, Vietnam
cuong.dq204902@sis.hust.edu.vn

Trung-Dung Nguyen
Ha Noi University of Science and Technology
Hanoi, Vietnam
dung.nt204906@sis.hust.edu.vn

Ha-Dat Mai
Ha Noi University of Science and Technology
Hanoi, Vietnam
dat.mh200135@sis.hust.edu.vn

Van-Duc Vu
FPT University
Hanoi, Vietnam
ducvvhe176438@fpt.edu.vn

Ngoc-Anh Hoang
FPT University
Hanoi, Vietnam
anhhnhe186401@fpt.edu.vn

Thu-Uyen Nguyen
FPT University
Hanoi, Vietnam
uyennthe176614@fpt.edu.vn

Van-Thiep Nguyen
FPT University
Hanoi, Vietnam
thiepnvhe173027@fpt.edu.vn

Phan Xuan Tan
College of Engineering, Shibaura Institute of Technology
Tokyo, Japan
tanpx@shibaura-it.ac.jp

Phong-Tung Doan (Corresponding author)
Ha Noi University of Science and Technology
Hanoi, Vietnam
tungdp@soict.hust.edu.vn

Dinh-Cuong Hoang
FPT University
Hanoi, Vietnam
cuonghd12@fe.edu.vn

ABSTRACT

In the dynamic landscape of industrial logistics, the precise identification of pallet locations is pivotal for efficient supply chain operations. Leveraging recent advancements in computer vision and the affordability of cameras, our research addresses the challenge of automated pallet detection in factory settings. We propose an innovative approach that combines synthetic data generation with domain adaptation techniques. The integration of synthetic and real-world unlabeled data, guided by domain adversarial training, overcomes challenges related to data scarcity and domain shift. Experiments demonstrate the effectiveness of our approach in classifying pallet states. The results showcase the model's adaptability to real-world variations, achieving impressive accuracy. Our work not only optimizes resource allocation in warehouses but also offers a blueprint for the seamless integration of Computer Vision into broader logistics applications. The transformative potential of our methodology is underscored by its efficiency, autonomy, and applicability across industries reliant on robust supply chain management.

CCS CONCEPTS

• **Computing methodologies** → **Computer vision**.

KEYWORDS

Domain Adaptation, Synthetic Data, DANN, Pallet Recognition, Pallet Dataset Creating

1 INTRODUCTION

In today's dynamic industrial landscape, logistics plays a pivotal role in ensuring smooth supply chain operations by orchestrating the seamless flow of goods from origin to destination. Fundamental tasks such as stacking, unloading, and intra-warehouse movements are essential in this process, with the precise identification of pallet locations being of utmost importance. Recent technological advancements, including the widespread availability of cost-effective cameras and the successful implementation of computer vision, have revolutionized logistics operations[6, 10, 14, 21, 24, 29]. The affordability of cameras has led to their widespread adoption in warehouses and distribution centers, while computer vision has enabled accurate identification and tracking of objects[2, 9, 11–13, 20, 22], particularly pallets, within the logistics environment.

However, as the industry increasingly explores the potential of Computer Vision for automated pallet detection in factory settings, a significant hurdle emerges the need for substantial real-world labeled data, presenting challenges in terms of both time and cost for data labeling. Some have turned to synthetic data generation as a solution[32][25][30], but this approach encounters its obstacle known as domain shift.

To tackle these challenges, we propose an innovative approach that combines synthetic data for training with domain adaptation techniques. We utilize Blender, an open-source 3D creation suite, for generating images and labels, and incorporate Domain-Adversarial Training of Neural Networks[7]. Our methodology comprises three key components: a pre-trained ResNet-34[8] serving as the feature extractor, a class classifier, and a domain classifier. The distinctive

feature of this architecture lies in the domain classifier, which plays a crucial role in enhancing the model's domain adaptation capabilities. Importantly, the implementation of the Domain-Adversarial Training of Neural Networks method necessitates a significant amount of real-world unlabeled data.

Building upon our methodology, we proceeded to conduct experiments to validate its effectiveness in a classification task, specifically discerning whether an image depicts an absence of a pallet, an empty pallet, or a pallet laden with goods. This classification holds significant operational implications. In cases where no pallet is detected, intervention is kept to a minimum. If only a pallet is identified, a smaller lifting device suffices. However, when goods are present on the pallet, a larger lifting device becomes imperative. Mastering this classification task promises substantial advantages in optimizing resource allocation within the warehouse.

The training phase involved an iterative process, wherein the synthetic data generated through Blender was seamlessly integrated with a carefully curated subset of real-world unlabeled data. This fusion of synthetic and real data not only addressed the scarcity of labeled samples but also facilitated the adaptation of the model to the specific nuances of the factory setting. The Domain-Adversarial Training of Neural Networks (DANN) technique played a pivotal role in bridging the gap between synthetic and real-world domains

The experimental results unequivocally showcase that the suggested approach delivers impressive performance and bears great potential for far-reaching applications within the field of automated logistics. Through the strategic integration of Deep Learning alongside domain adaptation techniques, we not only augment the effectiveness of existing operations but also lay the foundation for future breakthroughs in autonomous logistics systems. In summary, our research marks a substantial leap forward in addressing crucial challenges related to automated pallet detection in factory settings, concurrently optimizing financial resources for labeling data. Through the integration of synthetic data generation and domain adaptation, we've established a robust methodology that produces precise and dependable outcomes without the need for actual labeled data. This not only streamlines the effort, time, and financial investments typically associated with labeling data but also guarantees consistently positive results. The significance of this work transcends the warehouse, providing a comprehensive blueprint for the seamless integration of Computer Vision techniques into the broader logistics landscape.

2 RELATED WORKS

Image Classification

The evolution of image classification, a fundamental task in computer vision with applications from medical diagnosis to autonomous vehicles[23], has seen a significant shift in methodologies, datasets, and performance metrics over the years. Initial approaches to image classification relied heavily on handcrafted features and traditional machine learning algorithms, such as Histogram of Oriented Gradients (HOG)[4] and Scale-Invariant Feature Transform (SIFT) combined with classifiers like Support Vector Machines (SVM). While these methods showed promising results on certain tasks, they struggled with complex and diverse datasets. The landscape changed with the introduction of Convolutional Neural Networks (CNNs), with LeNet-5 being a pioneering example. Further advancements were seen with AlexNet[17], VGGNet[26], and GoogLeNet[27], which exhibited deeper architectures that significantly improved classification accuracy by extracting hierarchical features.

To combat the challenges of limited labeled data, the strategy of transfer learning and fine-tuning became prevalent. This involved using models pre-trained on extensive datasets like ImageNet[5] and fine-tuning them for specific tasks, leading to improved generalization capabilities and resource efficiency[3]. Attention mechanisms, particularly Self-Attention in Transformer models[28], also gained prominence, enabling models to focus on relevant image regions and enhancing the discriminative power of image representations. The field also saw an increased focus on adversarial defense and robustness, with techniques like adversarial training[18] and robust optimization aimed at tackling the vulnerability of image classification models to adversarial attacks.

The progression of image classification has been significantly shaped by the accessibility of extensive datasets like ImageNet, CIFAR-10, and COCO, which have become pivotal benchmarks for evaluating and comparing models. In assessing model performance, the field has moved beyond traditional accuracy metrics to embrace a diverse range of measures, including precision, recall, F1 score, and AUC-ROC. Our research incorporates some of these metrics to comprehensively evaluate the efficiency of our model. This shift in evaluation methods mirrors the broader transition from handcrafted features to advanced deep learning architectures. The continuous exploration of novel approaches underscores a commitment to pushing the limits of image classification performance, necessitating a nuanced understanding beyond basic accuracy metrics.

Domain Adaptation

Domain Adaptation is a machine learning and data science technique aimed at addressing a common challenge: when the distribution of data in the source domain, where a model is trained, differs from the distribution of data in the target domain, where the model is deployed or tested. In other words, it deals with situations where the model may not perform well on new, unseen data due to differences between the training and test data distributions.

Domain adaptation aims to bridge this gap by making the model more robust to domain shifts. It involves techniques that help the model generalize better to the target domain despite being trained on a different source domain.

Key concepts and methods in domain adaptation include:

Feature Adaptation: Modifying the input features to make them more domain-invariant. This can involve various techniques like feature selection, dimensionality reduction, or feature transformation.[15]

Instance Re-weighting: Giving different weights to instances from the source and target domains during training to reduce the impact of the domain shift.

Model Adaptation: Adjusting the model parameters to account for domain differences. Methods such as domain adversarial training or fine-tuning are often used.[31]

Transfer Learning: Utilizing knowledge learned from the source domain to improve performance on the target domain. This can include pre-trained models or shared representations.[19]

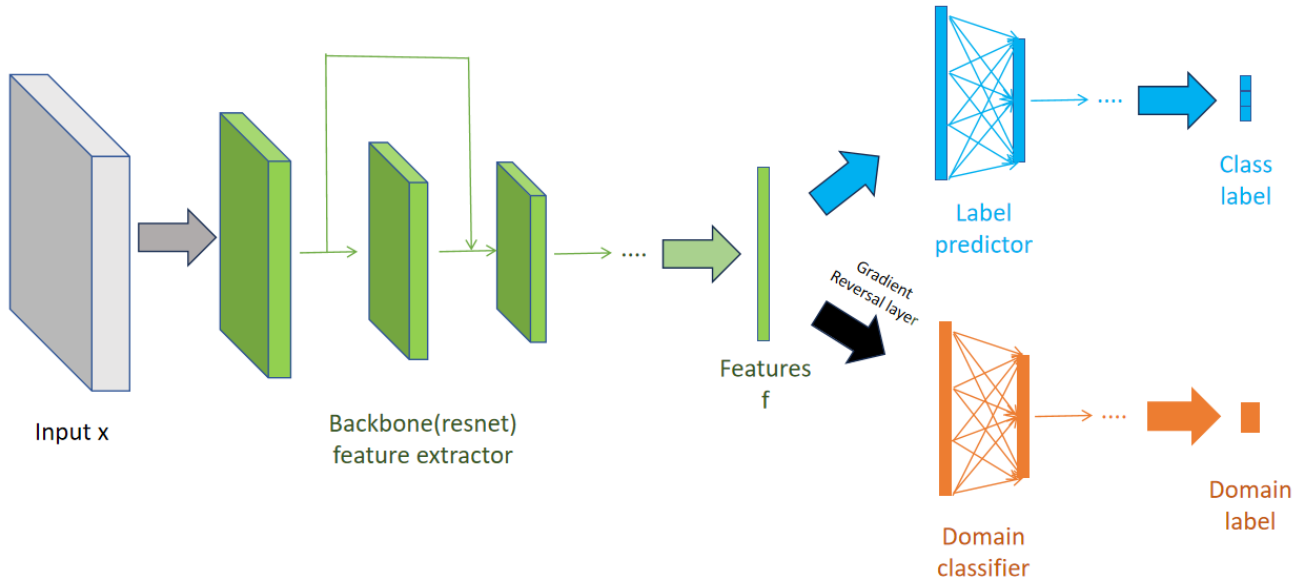

Figure 1: Model structure. The model contains 3 main components: The backbone block, the class predictor block, the domain classifier block.

Domain Shift Detection: Identifying when a domain shift occurs and adapting the model accordingly. Statistical tests and anomaly detection methods can be applied.

Domain adaptation is valuable in various real-world scenarios, such as sentiment analysis, image classification, and speech recognition, where the source and target domains may exhibit differences in terms of data distribution, data collection settings, or domain-specific factors. It plays a crucial role in ensuring that machine learning models perform effectively when deployed in practical applications where data can vary significantly.

Domain-Adversarial Training of Neural Networks

Top-performing deep architectures are trained on massive amounts of labeled data. In the absence of labeled data for a certain task, domain adaptation often provides an easy-to-get and cheap option given that labeled data of similar nature but from a different domain (e.g. synthetic images) are available. The Domain-Adversarial Training of Neural Networks paper proposed a new approach to domain adaptation in deep architectures known as DANN that can be trained on large amounts of labeled data from the source domain and large amounts of unlabeled data from the target domain (no labeled target domain data is necessary). As the training progresses, the approach promotes the emergence of âĂIJdeepâĂİ features that are discriminate for the main learning task on the source domain and invariant with respect to the shift between the domains. The authors show that this adaptation behaviour can be achieved in almost any feed-forward model by augmenting it with a few standard layers and a simple new gradient reversal layer. The resulting augmented architecture can be trained using standard back-propagation. Overall, the approach can be implemented with little effort using any of the deep-learning packages. The method performs very well in a series of image classification experiments, achieving an adaptation effect in the presence of big domain shifts and outperforming previous state-of-art on Office datasets.

The paper Incremental Unsupervised Domain-Adversarial Training of Neural Networks[1] focuses on incremental domain adaptation, where the model is adapted iteratively to the new domain by adding selected target samples to the source training set. This approach outperforms other state-of-the-art DA algorithms in various datasets. The paper makes use of DANN structure to identify the target samples on which there is greater confidence about their true label. The output of the model is analyzed in different ways to determine the candidate samples. The selected samples are then added to the source training set by self-labeling, and the process is repeated until all target samples are labeled. This approach implements a form of adversarial training in which, by moving the self-labeled samples from the target to the source set, the DA algorithm is forced to look for new features after each iteration.

3 METHODOLOGY

3.1 Model Building Overview

Building upon the successful domain-adversarial training framework (DANN), our approach tailors this architecture specifically for image classification tasks. We introduce targeted modifications to the DANN structure to effectively address the unique challenges and requirements of image classification. By leveraging the strengths of DANN while incorporating domain-specific adaptations, we aim to achieve superior performance in adapting models to new image classification domains. The model structure is presented in Figure 1.

In Figure 1, the employed model comprises key components: a backbone feature extractor, a layer for predicting labels, and a domain classifier layer integrated with a preceding gradient reversal layer. During each batch iteration, both real and synthetic data are fed into the model. This, coupled with the incorporation of a gradient reversal layer prior to the domain classifier, constitutes our adversarial training approach.

3.2 Backbone

ResNet is a convolution neural network (CNN) architecture for image classification. It was first introduced in the paper "Deep Residual Learning for Image Recognition" by He et al. (2016)[16] and has since become one of the most popular CNN architectures for a variety of tasks. The backbone we proposed for the model is pretrained ResNet-34. ResNet-34 consists of 34 convolutional layers, with each layer followed by a batch normalization layer and a ReLU activation function. The network also includes four "shortcut connections" that bypass groups of convolutional layers which are called residual blocks which are illustrated in Figure 2, the shortcut connection prevents vanishing gradients, enabling the network to learn long-range dependencies and deeper representations. These shortcut connections help to alleviate the vanishing gradient problem and allow the network to learn deeper representations.

3.3 Domain classifier and Label Predictor

Assume input is in m-dimensional real vector space, we have the space of the input $x = R^m$. Let $F_e(x, \theta_e)$ be the features map we get from the backbone. Also, we denote the label predictor layer and domain classifier layer output as $F_l(x, \theta_l)$ and $F_d(x, \theta_d)$ correspondingly. In the 3-layer output above, x is the input, with $\theta_e, \theta_l, \theta_d$ as the parameters of the corresponding layer. With y_i as the ground truth of the data point x_i, $L_l()$ as the loss function, we denoted the loss function for label and domain head:

$$L_l^i(\theta_e, \theta_l) = L_l(F_l(F_e(x_i, \theta_e), \theta_l), y_i) \tag{1}$$

$$L_d^i(\theta_e, \theta_d) = L_l(F_d(F_e(x_i, \theta_e), \theta_d), d_i) \tag{2}$$

To achieve adversarial training between the domain classifier and label predictor, we employ a reverse layer that inverts the gradient descent process from the domain classifier head by negating the calculated gradient. Subsequently, we train the model on both target and source data, but only the domain loss is calculated on the target data cause we assume the lack of labeled target data, as outlined in equation (3). This negative gradient descent from the domain head, coupled with losses from both source and target domains, is designed to prevent the model from discriminating between data originating from different domains. Consequently, the target domain label classifier is effectively trained using only labeled source data (synthetic data).

3.4 Data Feeding Mechanism

For each batch in an epoch, we will consecutively iterate through both the source dataset and target dataset, each will be denoted as S and T, respectively. For each iteration, the input data into the model will be X_i^S and X_i^T. After forwarding both data batches through the model, we will calculate the domain and label losses for both input data batches, and get Ls_l^i, Ls_d^i, Lt_d^i. We don't get label loss for the target data for archiving the tasks of training data image classification of target domain data with only labeled source(synthetic) domain data. The main Loss we use will be:

$$L_i = \lambda_1 . Ls_l^i + \lambda_2 . Ls_d^i + \lambda_3 . Lt_d^i \tag{3}$$

With each λ as the coefficients, we can experiment and choose the appropriate proportion of each loss. Also, the loss we use will be Cross-Entropy loss:

$$Loss = -\sum_{c=0}^{N-1} y_{i,c} . \log(p_{i,c}) \tag{4}$$

With N is the number of classes, y- binary indicator (0 or 1) if class label c is the correct classification for observation i, p is predicted probability observation i is of class c.

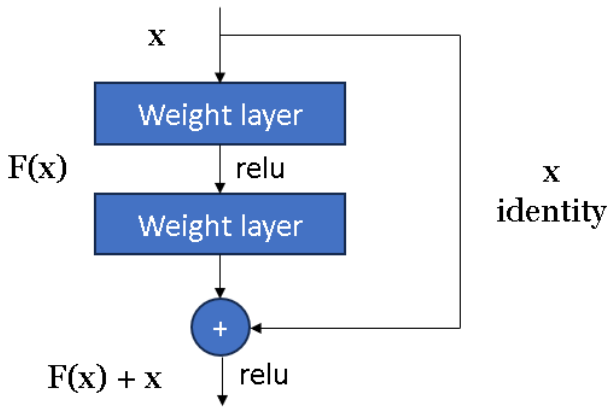

Figure 2: The residual block structure.

4 EXPERIMENTAL SETUP

4.1 Dataset

Regarding our dataset, we have meticulously annotated both real and synthetic images related to pallets. The dataset comprises three distinct labels: "No pallet" "Pallet only" and "Goods on pallet". The label "No pallet" indicates the absence of a pallet in the image,

suggesting either the presence of goods or an empty scene. "Pallet only" signifies the presence of a pallet in the image, without any accompanying goods. On the other hand, "Goods on pallet" denotes the presence of goods positioned on the pallet.

Our synthetic dataset, meticulously crafted using Blender, an open-source and versatile 3D graphics software, caters to diverse applications such as animation, visual effects, 3D printing, and video game development. Our focus lies in designing warehouse scenes within Blender, featuring key elements like pallets, goods, forklifts, and containers. To ensure comprehensive coverage, we implement camera movements and automate scene capture, yielding 1000 images per scene. Simultaneously, our Python scripting integrated into Blender handles automated labeling, encompassing tasks such as image classification, object detection, segmentation, and pose estimation. This automation taps into accurate object coordinates and orientations extracted from Blender, showcasing the software's adaptability.

Moreover, Blender's flexibility allows us to experiment with various lighting conditions, augmenting dataset diversity. Each image in our dataset is standardized to a size of (480,480), and the total number of images stands at an impressive 12,000, distributed evenly with 4000 images for each class.

The advantages of our automated labeling approach are particularly prominent in terms of time and cost efficiency, providing a viable alternative to the traditionally labor-intensive process of manually labeling real-world data for intricate computer vision tasks. Synthesizing datasets through Blender's 3D graphics and Python scripting not only streamlines image classification training but also emerges as a cost-effective and time-efficient solution for advanced computer vision applications.

Moving beyond image classification, our synthetic dataset creation process has successfully delved into object detection, a traditionally time-consuming task prone to manual errors. Leveraging Blender's 3D environment and Python scripting, we efficiently extract and incorporate precise object coordinates and types into the automated labeling process. This successful application in object detection underscores the versatility and effectiveness of our synthetic dataset generation approach, demonstrating its capacity to address complex computer vision tasks that typically demand substantial time and effort for manual labeling.

Regarding our dataset of real images, we have captured pallet-related videos with all three labels. For the labels "no pallet" and "goods on pallets," we have included various types of goods on the pallet to ensure dataset generalization. These videos are recorded at a resolution of 1440x1440 pixels, featuring a square image shape that facilitates the cropping data preprocessing step, making it both easier and more precise. The videos we've captured are recorded at 60 frames per second, with each video having a duration of approximately 50-60 seconds. However, we will not be including all frames in our dataset due to the high similarity of some frames, resulting from minor changes in pallet angles during recording. Therefore, for each label, we will select only approximately 1000 images. An illustrative example of our two datasets is shown in Figure 3.

Figure 3: The gallery of our synthetic and real datasets.

4.2 Setup

In this experimental study, we conducted a 30-epoch training using the Adam optimizer and ResNet34 pretrained on ImageNet1K. The final layer of the ResNet34 backbone generated a 512-dimensional latent representation, enriching the model's feature space. With a batch size of 16 and an initial learning rate of 0.001, we meticulously preprocessed input images by resizing them to (256,256) and applying a random crop to ensure standardization at (224,224). The model architecture, consisting of class and domain classifiers, incorporated a Fully Connected layer, ReLU activation function, and Dropout layer, fostering the learning of hierarchical representations. Post-training, a thorough evaluation of the model's performance informed hyperparameter fine-tuning, while data augmentation techniques were introduced to fortify the model's resilience. Our exploration included transfer learning strategies, fine-tuning for domain-specific challenges, and iterative adjustments to achieve optimal results. Simultaneously, rigorous testing on independent datasets verified the model's generalization and practical utility, concluding a systematic and refined approach to model development.

4.3 Experimental results

Data Augmentation Strategies: During our experimentation, we explored several data augmentation strategies, including mosaic, cutmix, mixup, and simpler techniques like image flipping and rotation. The results revealed that the complex augmentation techniques, such as mosaic, cutmix, and mixup, did not yield significant benefits and, in some instances, led to a reduction in performance. With preprocessing techniques in this section, we delve into the systematic examination of preprocessing techniques applied to our automated pallet classification model. The insights garnered from this analysis are pivotal in understanding the influence of preprocessing on the model's performance. The application of preprocessing techniques significantly influenced the model's performance. The normalization step, is implemented through

Table 1: Score of our models

Method	F1 score	Precision	Recall	Accuracy
Ours(train by synthetic, without DANN)	0.81	0.85	0.77	0.79
Ours(train by real, without DANN)	1	1	1	1
Ours(DANN without augmentation)	0.77	0.78	0.75	0.76
Ours(DANN, ResNet-18)	0.84	0.85	0.82	0.84
Ours(DANN, ResNet-34)	0.99	0.99	0.99	0.99

transforms.Normalize(mean=(0.485, 0.456, 0.406), std=(0.229, 0.224, 0.225)), and the use of transforms.Ran- domCrop(224) played a pivotal role in surpassing the 0.9 accuracy threshold. Normalizing input images helped the model better adapt to variations in lighting conditions and color distributions, leading to improved overall performance. Employing random cropping at the size of 224x224 pixels contributed to spatial feature extraction, enhancing the model's ability to discern critical details. Model Sensitivity: Comparing different backbone architectures, we observed that ResNet34 consistently outperformed ResNet18 in terms of accuracy. The deeper architecture of ResNet34 provided a more expressive feature representation, enabling the model to capture intricate details and patterns more effectively. This finding emphasizes the importance of selecting a suitable backbone architecture tailored to the complexity of the classification task. The experiment "Train by synthetic, without DANN" illustrated the model's transferability from synthetic to real data, achieving stable performance on the test set. In contrast, "Train=real, without DANN" demonstrated that the model reached maximum accuracy when trained and tested on the same real-world data source. The results of "DANN without augmentation" and "DANN, ResNet-18" continued to emphasize the role of DANN in reducing the domain gap between the two data domains. Ultimately, Ours yielded the most impressive performance, with metrics approaching their maximum values. The flexible integration of augmentation, DANN, and the ResNet-34 model architecture produced a robust model, showcasing the transferability and effectiveness of the proposed method in the real-world pallet classification task. The experimental results underscore the effectiveness of our proposed methodology in tackling the automated pallet detection challenge. On the source dataset, our model achieved an outstanding accuracy of 0.99, showcasing its capability to accurately classify pallet states within a controlled environment. Moreover, on the target dataset, our model demonstrated a commendable accuracy of 0.99. This performance on the target dataset signifies the success of our domain adaptation techniques in ensuring the model's adaptability to real-world variations and shifts in data distribution. The summary of our results is written in Table 1: Score of our models

5 CONCLUSION

In conclusion, our research presents a pioneering solution to the automated pallet detection challenge within factory settings. By synergizing synthetic data generation through Blender with domain adaptation techniques, we have established a robust methodology capable of delivering accurate and reliable results. This integrated approach not only addresses the scarcity of labeled samples but also facilitates model adaptation to the nuanced realities of factory environments. The potential impact of our work extends beyond warehouse confines, offering a comprehensive blueprint for seamlessly integrating Computer Vision techniques into the broader logistics landscape. As we envision the future of autonomous logistics systems, our proposed methodology stands as a testament to the transformative power of combining cutting-edge technologies for real-world problem-solving. The success of our approach signifies a promising direction for enhancing the efficiency and autonomy of logistics operations, with implications for diverse industries reliant on seamless supply chain management.

REFERENCES

[1] Jorge Calvo-Zaragoza Antonio-Javier Gallego and Robert B. Fisher. 2020 IEEE. Incremental Unsupervised Domain-Adversarial Training of Neural Networks. (2020 IEEE).

[2] Srikanth Bethu, M. Neelakantappa, A. Swami Goud, B. Hari Krishna, and P. N. V. Syamala Rao M. 2023. An Approach for Person Detection along with Object Using Machine Learning. *Journal of Advances in Information Technology* 14, 3 (2023), 411–417.

[3] Wuyang Chen, Zhiding Yu, Shalini De Mello, Sifei Liu, Jose M Alvarez, Zhangyang Wang, and Anima Anandkumar. 2021. Contrastive syn-to-real generalization. *arXiv preprint arXiv:2104.02290* (2021).

[4] Navneet Dalal and Bill Triggs. 2005. Histograms of oriented gradients for human detection. 1 (2005), 886–893.

[5] Jia Deng. 2009. A large-scale hierarchical image database. *Proc. of IEEE Computer Vision and Pattern Recognition, 2009* (2009).

[6] M Di Capua, A Ciaramella, and A De Prisco. 2023. Machine learning and computer vision for the automation of processes in advanced logistics: The integrated logistic platform (ILP) 4.0. *Procedia Computer Science* 217 (2023), 326–338.

[7] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, and Germain. 2016. Domain-adversarial training of neural networks. *The journal of machine learning research* 17, 1 (2016), 2096–2030.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. (2016), 770–778.

[9] Dinh-Cuong Hoang, Liang-Chia Chen, and Thanh-Hung Nguyen. 2016. Sub-OBB based object recognition and localization algorithm using range images. *Measurement Science and Technology* 28, 2 (2016), 025401.

[10] Dinh-Cuong Hoang, Achim J Lilienthal, and Todor Stoyanov. 2020. Object-RPE: Dense 3D reconstruction and pose estimation with convolutional neural networks. *Robotics and Autonomous Systems* 133 (2020), 103632.

[11] Dinh-Cuong Hoang, Anh-Nhat Nguyen, Van-Duc Vu, Duy-Quang Vu, Van-Thiep Nguyen, Thu-Uyen Nguyen, Cong-Trinh Tran, Khanh-Toan Phan, and Ngoc-Trung Ho. 2023. Grasp Configuration Synthesis from 3D Point Clouds with Attention Mechanism. *Journal of Intelligent & Robotic Systems* 109, 3 (2023), 71.

[12] Dinh-Cuong Hoang, Johannes A Stork, and Todor Stoyanov. 2022. Context-aware grasp generation in cluttered scenes. In *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 1492–1498.

[13] Dinh-Cuong Hoang, Johannes A Stork, and Todor Stoyanov. 2022. Voting and Attention-Based Pose Relation Learning for Object Pose Estimation From 3D Point Clouds. *IEEE Robotics and Automation Letters* 7, 4 (2022), 8980–8987.

[14] Dinh-Cuong Hoang, Todor Stoyanov, and Achim J Lilienthal. 2019. Object-rpe: Dense 3d reconstruction and pose estimation with convolutional neural networks for warehouse robots. In *2019 European Conference on Mobile Robots (ECMR)*. IEEE, 1–6.

[15] Anne Marie Amja; Abdel Obaid; Hafedh Mili; Zahi Jarir. 2016", journal = 2016 IEEE 2nd International Conference on Collaboration and Internet Computing (CIC),. Feature-Based Adaptation and Its Implementation. (2016", journal = 2016 IEEE 2nd International Conference on Collaboration and Internet Computing (CIC),).

[16] Shaoqing Ren Jian Sun Kaiming He, Xiangyu Zhang. 2015. Deep Residual Learning for Image Recognition. (2015).

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* 25 (2012).

[18] Gabriel Resende Machado, Eugenio Silva, and Ronaldo Ribeiro Goldschmidt. 2021. Adversarial machine learning in image classification: A survey toward the defenderâĂŹs perspective. *ACM Computing Surveys (CSUR)* 55, 1 (2021), 1–38.

[19] Jianmin Wang Michael I. Jordan Mingsheng Long, Yue Cao. 2015. Learning Transferable Features with Deep Adaptation Networks. (2015).

[20] Adedeji Olugboja, Zenghui Wang, and Yanxia Sun. 2021. Parallel Convolutional Neural Networks for Object Detection. *Journal of Advances in Information Technology* 12, 4 (November 2021), 279–286. https://doi.org/10.12720/jait.12.4.279-286

[21] Alessandro Palleschi, Marco Gugliotta, Chiara Gabellieri, Dinh-Cuong Hoang, Todor Stoyanov, Manolo Garabini, and Lucia Pallottino. 2020. Fully autonomous picking with a dual-arm platform for intralogistics. In *Proc. I-RIM Conf. I-RIM*. 109–111.

[22] Luyl-Da Quach, Khang Nguyen Quoc, Anh Nguyen Quynh, and Hoang Tran Ngoc. 2023. Evaluating the Effectiveness of YOLO Models in Different Sized Object Detection and Feature-Based Classification of Small Objects. *Journal of Advances in Information Technology* 14, 5 (2023), 907–917.

[23] R Joshua Samuel Raj, S Jeya Shobana, Irina Valeryevna Pustokhina, Denis Alexandrovich Pustokhin, Deepak Gupta, and KJIA Shankar. 2020. Optimal feature selection-based medical image classification using deep learning model in internet of medical things. *IEEE Access* 8 (2020), 58006–58017.

[24] Jerome Rutinowski, Hazem Youssef, Anas Gouda, Christopher Reining, and Moritz Roidl. 2022. The Potential of Deep Learning based Computer Vision in Warehousing Logistics. *Logistics Journal: Proceedings* 2022, 18 (2022).

[25] Jacob Shermeyer, Thomas Hossler, Adam Van Etten, Daniel Hogan, Ryan Lewis, and Daeil Kim. 2021. RarePlanes: Synthetic Data Takes Flight. (January 2021), 207–217.

[26] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. (2015). arXiv:cs.CV/1409.1556

[27] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. (2015), 1–9.

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[29] Van-Duc Vu, Dinh-Dai Hoang, Phan Xuan Tan, Van-Thiep Nguyen, Thu-Uyen Nguyen, Ngoc-Anh Hoang, Khanh-Toan Phan, Duc-Thanh Tran, Duy-Quang Vu, Phuc-Quan Ngo, et al. 2024. Occlusion-Robust Pallet Pose Estimation for Warehouse Automation. *IEEE Access* (2024).

[30] Erroll Wood, Tadas Baltrusaitis, Charlie Hewitt, Sebastian Dziadzio, Thomas J. Cashman, and Jamie Shotton. 2021. Fake It Till You Make It: Face Analysis in the Wild Using Synthetic Data Alone. (October 2021), 3681–3691.

[31] Hana Ajakan Pascal Germain Hugo Larochelle FranÃğois Laviolette Mario Marchand Victor Lempitsky Yaroslav Ganin, Evgeniya Ustinova. 2015. Domain-Adversarial Training of Neural Networks. (2015).

[32] Yang Zheng, Adam W. Harley, Bokui Shen, Gordon Wetzstein, and Leonidas J. Guibas. 2023. PointOdyssey: A Large-Scale Synthetic Dataset for Long-Term Point Tracking. (October 2023), 19855–19865.