Doctoral School of Applied Informatics and Applied Mathematics

Óbuda University

New Deep Neural Network Applications in Robot Control and System Supervision

Ph.D. Thesis Summary

Artúr István Károly

Supervisor:
Dr. Péter Galambos

Budapest
2023

Motivation

Machine learning solutions are gradually making their presence felt across a wide range of industries, subtly impacting our daily lives, professional services, and military applications. The manufacturing industry, in particular, benefits significantly from these advancements, with numerous research projects exploring the potential of Industry 4.0 concepts with great fervor.

Out of all machine learning approaches, Deep learning (DL) incorporates some of the most significant and impactful improvements of recent years. It has introduced innovative advancements in various aspects of robotics throughout its development as well [1]. Despite the fact that DL-based solutions can be applied to a diverse range of problems, they have the disadvantages of unpredictability and high computational complexity [2, 3]. In safety-critical systems, like self-driving cars and industrial robots, DL methods are never used independently, and their output is always treated with uncertainty. As a result, these DL methods are often tested on benchmarks that assess their robustness [4, 5, 6, 7]. Due to the high computational complexity and time-consuming training process of DL systems, alternative model architectures and training strategies have been introduced, such as deep convolutional neural networks [8, 9] and transfer learning [10, 11, 12], to enable the training of robust models with limited resources.

In addition to the training process, data collection and preparation for DL require significant resources as well, especially if done manually [3]. This issue is particularly relevant in robotics, where data collection often involves performing actions on an actual robot [13]. Such data collection can take months of robot hours and require multiple robots, resulting in significant costs. To minimize the amount of labeled data required for training, new DL approaches are utilizing unsupervised/semi-supervised methods and transfer learning [14, 15]. Moreover, some approaches aim to reduce resource requirements by collecting data primarily in simulation instead of reality [16, 17, 18].

This work focuses mainly on perception-level problems, including object detection, segmentation, and other related tasks essential for robotic manipulation and mobile robot navigation. While previous studies have indicated that DL methods can provide promising results, they also demonstrate significant limitations caused by high computational complexity and the requirement of extensive training data. Given the importance of resource efficiency in robotics, searching for improved DL-based solutions to overcome these challenges becomes crucial. To address these issues, this work explores unsupervised learning techniques, transfer learning, and automated dataset-generation methodologies for robotics.

Thesis 1

State and anomaly detection based on real-time clustering

Preliminaries

The Support Vector Machine (SVM) method trains a linear classifier for binary classification using a decision function of the form

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b,$$

where $\mathbf{x}$ is the input vector, $\mathbf{w}$ is the weight vector and b is the bias [19]. Predictions are made based on the values of $f(\mathbf{x})$, where $f(\mathbf{x}) \geq 0$ results in a prediction of $y = 1$, and $f(\mathbf{x}) < 0$ results in a prediction of $y = -1$ [19]. While standard SVMs determine the parameters of the decision function using a training dataset with ground-truth labels, the One-Class Support Vector Machine (OCSVM) is specifically designed to distinguish samples belonging to one class from those of any other class [20, 21]. As a result, unsupervised training is possible using a set of unlabeled samples that are assumed to belong to the same class, which is useful for anomaly detection when rare data must be distinguished from the rest.

When implementing OCSVMs, there are typically two approaches that are used. One method, as outlined by Schölkopf et al. [20], involves separating the training samples from the origin using a hyperplane in the feature space and maximizing the distance between the hyperplane and the origin.

The second method, introduced by Tax and Duin [21] involves enclosing the training samples in the feature space with a spherical surface and minimizing the volume of this hypersphere.

The decision function that classifies a given sample $\mathbf{x}$ as a member of the class is obtained by solving the maximization or minimization problem using the Lagrange multiplier method (utilizing specialized solver algorithms such as SMO [22, 23]). Distances in the feature space can also be computed using the kernel method [21], which transforms the problem to a higher-dimensional space where the samples become linearly separable.

The Radial Basis Function (RBF), also known as the Gaussian kernel, is the most frequently used kernel function. Its formulation for two data points, $\mathbf{x}_i$ and $\mathbf{x}_j$, is given by

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right),$$

where the kernel parameter σ controls the sensitivity of the kernel function and should be set to a suitable empirically chosen value.

The series expansion of the RBF kernel function leads to an infinite series, in which the terms

$$\langle \mathbf{x}_i, \mathbf{x}_j \rangle, \langle \mathbf{x}_i, \mathbf{x}_j \rangle^2, \langle \mathbf{x}_i, \mathbf{x}_j \rangle^3 \dots$$

are present, which are also kernel functions on their own. This provides the flexibility to design the classifier in a space with arbitrary dimensionality, allowing for nonlinear decision boundaries to be formed in the feature space.

Building on the theory of unsupervised learning with OCSVMs, I proposed an unsupervised real-time algorithm (Algorithm 1.) that dynamically trains an ensemble ($\mathbf{E}$) of OCSVM models for automatically discovering the state of robot applications and for the detection of anomalies. The algorithm operates on a data stream $\mathcal{S}$. In $\mathcal{S}$, data points $\mathbf{X}_i | i \in \{t, t-1, t-2, ...\}$ are available at a sampling rate (f Hz) defined by the robot controller setup, with $\mathbf{X}_t$ being the most recent data point at time step t. For the OCSVM inference and training, sliding window sampling is used. The inference is performed on overlapping sliding windows ($\mathbf{W}^t$) which ensures real-time behavior. On the other hand, the training set ($\mathbf{T}$) consists of non-overlapping samples ($\mathbf{W}^t_{train}$) to avoid overfitting and to reduce computational costs. The algorithm can be parameterized with w and n, which dictate the widths of the sliding windows and the number of samples used for training, respectively. I provide formulas to determine the algorithm's computational complexity for a given set of parameters using linear or non-linear (RBF) kernels.

The algorithm creates a contingency table ($\mathbf{C}$) that indicates how often different OCSVMs in the ensemble "fired" together. This information is then used to form groups of OCSVMs with the aim of building hierarchies of recognized states (Algorithm 2). The hierarchy building is performed offline, on a recorded segment of the data stream ($\mathcal{R}$), to guarantee that all OCSVMs will receive the same inputs.

I also showed that the state discovery algorithm can be used for evaluating and comparing generative machine learning models, such as Generative Adversarial Neural Networks (GANs) by analyzing their synthesized outputs. The experimental results suggest that this method could be a great alternative in use cases where for the synthesized data semantics pre-trained feature extractors are not available. However, the algorithm in itself is unable to identify mode collapse and should be used in combination with a diversity measure.

New Scientific Results

Thesis 1

I present a new clustering algorithm (Algorithm 1.) for the automatic online and real-time classification of operation states and detection of anomalies in robotic applications utilizing finite state descriptor dimensions and continuous numerical features. Besides robotics applications, the use of the algorithm can be generalized to the evaluation of generative machine learning models. The effectiveness of the proposed method was demonstrated on a representative collaborative robot application, as well as through successful implementation in a real-world industrial setting.

Algorithm 1: Unsupervised clustering algorithm

input: $\mathcal{S}$, w, n, *stopping_criterion*, *cd*

```
/* Initialize internal variables                                    */
W^t = [], W^t_train = [], T = [] E = [];
C = [[]];
i = 0, count = 0;
stop = False, no_train_step = 0;
on new X_t in S:
    /* Update data structures                                       */
    i += 1;
    W^t.append(X_t);
    if W^t.size() > w then
        W^t.remove(0);
    if i == w then
        W^t_train=W^t;
        i = 0;
        T.append(W^t_train);
        if T.size() > n then
            T.remove(0);
    else
        W^t_train=W^{t-1}_train;
    /* Perform predictions                                          */
    p = [];
    for OCSVM in E do
        p.append(OCSVM.predict(W^t));
    /* Train a new OCSVM if needed                                  */
    if !stop and (all(p==-1) or E.size() == 0) and T.size() == n then
        if count < cd then
            count += 1;
        else
            E.append(OCSVM.train(T));
            count = 0;
    else
        no_train_step += 1;
        if no_train_step == stopping_criterion then
            stop = True;
    /* Update contingency table                                     */
    C_update(p, C);
```

Sub-thesis 1.1

I showed that a dynamically constructed ensemble of OCSVMs together with a contingency table can be used to discover a multi-level hierarchy of elementary states using a bottom-up hierarchy-building strategy (Algorithm 2.).

Algorithm 2: Bottom-up hierarchy building strategy

```
input : E, C, R, th
/* Initialize internal variables                                    */
H = [], G = [];
N = R.size();
/* Calculate entropies                                              */
for OCSVM in E do
    H.append(Entropy(OCSVM.predict(R)));
H /= max(H);
for h in H do
    /* Find the index of the minimal entropy OCSVM                  */
    i_h = argmin(H);
    if i_h in G then
        H[i_h] = 2 ;                  // OCSVM i_h is already in a group
    else
        /* Create a new group                                       */
        G.append([i_h]);
        I = [];
        for j = 0; j < C[i_h].size(), j++ do
            I.append(Info_Gain(C,(i_h, j), N)); // Compute Information Gain
              for i_h
        I[i_h] = -1;
        for j = 0; j < I.size(); j++ do
            /* Start with the most similar OCSVM                    */
            i_g = argmax(I/max(I));
            if C[i_h][i_g]/C[i_h][i_h] ¿ th then
                G[-1].append(i_g) ;       // Add i_g to the current group
                I[i_g] = -1;
            else
                break;
        H[i_h] = 2;
return: G
```

Sub-thesis 1.2

Through representative examples, I showed that using non-overlapping sliding windows in the input data stream for acquiring training samples significantly reduces the computational time without significantly degrading the prediction performance. I provided formulas for determining the computational requirements depending on the parameters of the method (1) and (2).

$$t_{train} \propto N^{OCSVM} w \mathcal{O}_{train}$$

linear:

$$\mathcal{O}_{train} = O(dn) \rightarrow t_{train} \propto N^{OCSVM} wdn \approx N^{OCSVM} \mathcal{T} fd \tag{1}$$

non-linear:

$$\mathcal{O}_{train} = O(dn^2) \rightarrow t_{train} \propto N^{OCSVM} wdn^2 \approx N^{OCSVM} \mathcal{T} fdn$$

where $\mathcal{O}_{train}$ is the original computational complexity of training an OCSVM directly on the data points.

The computational complexity for the inference with our algorithm is

$$t_{inference} \propto N^{OCSVM} w \mathcal{O}_{inference}$$

linear:

$$\mathcal{O}_{inference} = O(d) \rightarrow t_{inference} \propto N^{OCSVM} wd \tag{2}$$

non-linear:

$$\mathcal{O}_{inference} = O(dn) \rightarrow t_{inference} \propto N^{OCSVM} wdn \approx N^{OCSVM} \mathcal{T} fd$$

where $\mathcal{O}_{inference}$ is the original computational complexity of predicting with an OCSVM directly on the data points.

Sub-thesis 1.3

I demonstrated that the OCSVM-based anomaly detection approach, initially designed for robotics applications, can be effectively used to evaluate generative machine learning models through statistical analysis of synthesized outputs. Unlike current evaluation methods, this approach is able to evaluate models independently of the output data semantics, as it does not require a pre-trained feature extractor.

Corresponding publications: [KA1, KA2, KA3].

Thesis 2
Cross-modal mapping-based transfer learning using pre-trained RGB feature extractors

Preliminaries

In many vision-based robotics solutions, such as moving obstacle detection for mobile robotics, incorporating additional modalities on top of the typically used RGB features is often beneficial [24, 25]. The most commonly utilized vision-related non-RGB modalities are depth, surface normals, and optical flow. Since these modalities are closely related to vision, they all have corresponding visual representations/interpretations, which are generally used for visualization purposes. Figure 1. shows some examples of such representations.

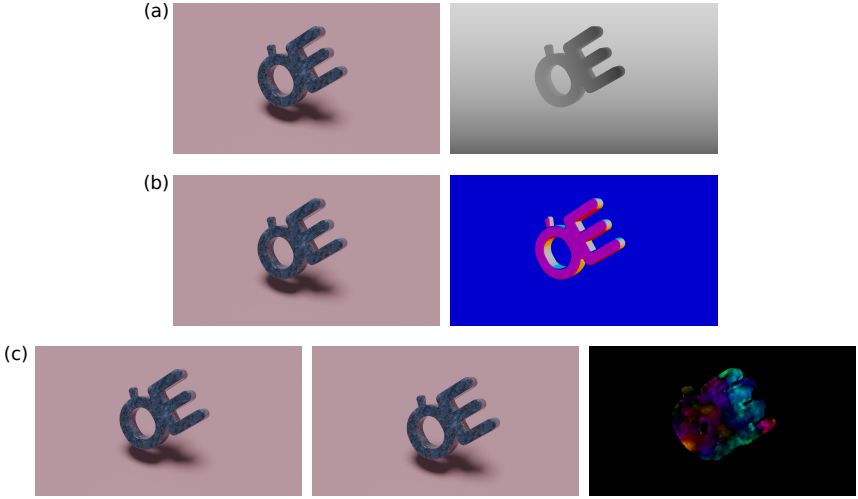

Figure 1: Image representation of non-RGB modalities. **a**: RGB image and corresponding depth data as a grayscale image, **b**: RGB image and corresponding surface normals as RGB image, **c**: Consecutive RGB frames and corresponding optical flow as RGB image (using the 2D polar color map method)

Acquiring a publicly available dataset for segmenting moving objects in mobile robotics, tailored to a specific task, can pose a significant challenge, often necessitating the development of a customized training dataset for each specific use

case. In order to decrease the required number of training samples, pre-trained feature extractors can be utilized using transfer learning [12, 11]. However, the most commonly used pre-trained feature extractors are trained using large-scale RGB image datasets and thus cannot directly process other modalities.

My proposal is that with the correct formulation of the inputs, feature extractors that were previously pre-trained on RGB images can be repurposed to process non-RGB modalities. This input formulation is called cross-modal mapping.

In order to prove this hypothesis, a new deep neural network was proposed called the Optical Flow Segmentation Network (OFSNet) for moving object segmentation in video sequences for mobile robot navigation. The OFSNet model is based on the popular U-Net architecture [26] and uses the Inception v3 feature extractor [27] (Table 1).

Table 1: The structure of the OFSNet model, including the Inception v3 feature extractor from #1 to #12. The network structure from #13 to #18 is our contribution, and only the parameters of this part were modified during the training process.

#	type	**patch size/stride** or remarks	**input size**
Layers from Inception v3 model			
1	conv	3x3/2	299x299x3
2	conv	3x3/1	149x149x32
3	conv padded	3x3/1	147x147x32
4	pool	3x3/2	147x147x64
5	conv	3x3/1	73x73x64
6	conv	3x3/2	71x71x80
7	conv	3x3/1	35x35x192
8	3 x Inception	Inception block	35x35x288
9	5 x Inception	Inception block	17x17x768
10	2 x Inception	Inception block	8x8x1280
11	pool	8x8	8x8x2048
12	linear	Inception v3 features	1x1x2048
Layers for segmentation			
13	transposed conv	3x3/2	1x1x2048
14	transposed conv	4x4/2	3x3x1280
15	skip connection	#9+#14	8x8x1280
16	transposed conv	16x16/2	8x8x1280
17	linear	logits	30x30x1
18	sigmoid	classifier	30x30x1

The training of the OFSNet model was performed in a self-supervised fashion, using the Unsupervised Non-Local Consensus Voting (uNLC) method [28] for generating ground-truth segmentation masks.

The moving objects often appear relatively small in the images, leading to a class imbalance regarding the number of positive and negative pixels. This imbalance can hinder the convergence of the training process by affecting the value of the computed loss and, thus, the magnitude of the gradients as well.

The Cross-Entropy loss is a typical loss for segmentation models

$$L_{CE} = -\frac{1}{N}\sum_{i=1}^{N} y_i log_2(p_i) + (1-y_i) log_2(1-p_i),$$

where L_{CE} is the Cross-Entropy loss for a single frame, and N represents the total number of pixels in the output. The correct label for each pixel is represented by y_i, where a value of 0 denotes the background, and a value of 1 denotes the moving object. The predicted probability of the i^{th} pixel belonging to the moving object is represented by p_i.

Another commonly used loss function is the Soft Dice loss [29, 30]

$$\begin{aligned} SoftDice &= \frac{2\sum_{i=1}^{N} y_i p_i}{\sum_{i=1}^{N} p_i + \sum_{i=1}^{N} y_i}, \\ L_{SD} &= 1 - SoftDice. \end{aligned} \tag{3}$$

where L_{SD} is the Soft Dice loss, N denotes the total number of pixels in the output, while y_i represents the accurate label for the i^{th} output pixel. Values of 0 represent the background, and a value of 1 represents the foreground, which corresponds to the moving object. The predicted probability of the i^{th} output pixel belonging to the moving object is represented by p_i.

Since the Cross-Entropy loss penalizes false-positive predictions more, and the Soft Dice Loss penalizes false negatives more, I proposed a new loss function, the compound loss, to overcome the class imbalance in the training samples. The compound loss uses a dynamically adjustable linear combination of the Cross-Entropy and the Soft Dice loss.

New Scientific Results

Thesis 2
I developed a Deep Learning model (OFSNet), and a corresponding loss function for moving object segmentation in video sequences, enabling moving obstacle avoidance in indoor environments for mobile robot navigation. The proposed method has been validated in a real-world industrial Automated Guided Vehicle (AGV) system prototype.

Sub-thesis 2.1
I demonstrated that feature extractors pre-trained on real-world RGB images can generalize to combined optical flow and grayscale input data with appropriate formatting/cross-modal mapping (4). This conclusion was supported by experiments on the DAVIS 2016 dataset and real-world data acquired from an industrial Automated Guided Vehicle (AGV) system prototype.

$$\begin{aligned}
\hat{R} &= \mathcal{F}_{::1} + abs(min(\mathcal{F}_{::1}))\mathbf{J}^{w\times h}) \\
\hat{G} &= \mathcal{F}_{::2} + abs(min(\mathcal{F}_{::2}))\mathbf{J}^{w\times h}) \\
\mathcal{I}^{RGB}_{::1} &= \frac{\hat{R}}{max(\hat{R})} \\
\mathcal{I}^{RGB}_{::2} &= \frac{\hat{G}}{max(\hat{G})} \\
\mathcal{I}^{RGB}_{::3} &= \frac{\mathcal{I}^{Y}}{max(\mathcal{I}^{Y})}
\end{aligned} \tag{4}$$

Sub-thesis 2.2
I introduced a compound loss function (5) and a corresponding empirical training approach that utilizes a dynamic linear combination of Cross-Entropy Loss and Soft Dice Loss functions to overcome their counter-effecting biases. The effectiveness of this loss function and training strategy was demonstrated through the training of the OFSNet model. I demonstrated that the weighting parameter (α) can be dynamically adjusted to ensure the best overlap between the ground truth and the predicted segmentation masks, even in unbalanced (in terms of object-to-background ratio) datasets.

$$\begin{aligned}
L &= (1-\alpha)L_{CE} + \alpha L_{SD} \\
\alpha &= \frac{\left(\frac{j}{n_e}\right)^4}{1.6}
\end{aligned} \tag{5}$$

Corresponding publications: [KA4, KA5, KA6].

Thesis 3

Automatic large-scale visual dataset generation

Preliminaries

Dataset preparation and data annotation are significant roadblocks to developing new Deep Learning (DL) solutions. With synthetic data, the collection and annotation procedure can be easily automated. However, additional measures must be taken to ensure that the models trained on the synthetic data can adapt to the real-world domain [16, 17, 18]. This problem is often referred to as "bridging the reality gap". In the case of real-world samples, however, there is no problem with model adaptation, but we cannot take advantage of automated annotation, which presents a challenge. I propose two methods for automated dataset generation for visual robotics challenges.

The first method enables automated annotation for object segmentation in real images, specifically tailored for robotics applications. The method leverages the unique features of robotics, such as access to the robot's pose information and the ability to mount a camera on the robot. Additionally, it takes advantage of the fact that objects with known geometries can be placed in predetermined poses within the robot's workspace. Under these assumptions, we can describe the digital twin of the whole scene with a high degree of accuracy, enabling us to compute virtual camera projections. The virtually projected information augments the actual photos with annotations that make the training dataset. This approach extends the digital twin paradigm to the field of DL dataset creation and validation.

The generation of instance segmentation masks is accomplished by computing the perspective projection of 3D points located on the surfaces of the objects onto the image plane. The perspective projection $\bar{\mathbf{x}} = (u, v, 1)^\top$ of a 3D point ${}^w\mathbf{X} = ({}^wX, {}^wY, {}^wZ, 1)^\top$ (given in the world frame) is described by

$$\bar{\mathbf{x}} = \mathbf{K}\mathbf{\Pi}\ {}^c\mathbf{T}_w\ {}^w\mathbf{X}, \tag{6}$$

where $\mathbf{K}$ represents the camera matrix, which contains the intrinsic parameters of the camera and can be determined by camera calibration [31]. The projection matrix $\mathbf{\Pi}$ is in the form of $[\mathbf{I}|\mathbf{0}]$, where $\mathbf{I}$ is a 3×3 identity matrix and $\mathbf{0}$ is a column vector of three zeros. Finally, ${}^c\mathbf{T}_w$ is the 4×4 homogeneous transformation matrix that describes the transformation between the world and the camera frame.

Algorithm 3. describes the proposed method. For the symbolic description of the problem, let $P({}^{w}\mathbf{X})$ denote the perspective projection of the point ${}^{w}\mathbf{X}$, F a face defined by a set of points ($F = \{{}^{w}\mathbf{X}_1, {}^{w}\mathbf{X}_2, ..., {}^{w}\mathbf{X}_n\}$), $R({}^{w}\mathbf{X})$ a ray coming from the origin of the camera frame and going through the point ${}^{w}\mathbf{X}$, and ${}^{w}\mathbf{X}^{\mathcal{O}}_{all}$ all the possible points on the surface of object $\mathcal{O}$. To create segmentation masks, a finite set of points on the surface of each object needs to be selected: ${}^{w}\mathbf{X}^{\mathcal{O}} = \{{}^{w}\mathbf{X} | {}^{w}\mathbf{X} \text{ on the surface of } \mathcal{O}\}$, ${}^{w}\mathbf{X}^{\mathcal{O}} \subseteq {}^{w}\mathbf{X}^{\mathcal{O}}_{all}$. The power set $\mathbb{P}({}^{w}\mathbf{X}^{\mathcal{O}})$ contains all the possible (not necessarily meaningful) faces for object $\mathcal{O}$, for a given set of surface points ${}^{w}\mathbf{X}^{\mathcal{O}}$. Polygons can be formed in the image plane by projecting each point of a face: $Poly^{F} = \{P({}^{w}\mathbf{X}_i) \text{ for } {}^{w}\mathbf{X}_i \in F\}$, and using the projections as the vertices of the polygon. A set of faces $F^{\mathcal{O}} \subseteq \mathbb{P}({}^{w}\mathbf{X}^{\mathcal{O}})$ have to be chosen for object $\mathcal{O}$, such that all the projections given by $P({}^{w}\mathbf{X}_j)$ for ${}^{w}\mathbf{X}_j \in {}^{w}\mathbf{X}^{\mathcal{O}}_{all}$ fall inside at least one of the polygons of $Poly^{F_k}$, for $F_k \in F^{\mathcal{O}}$, but projections $P({}^{w}\mathbf{X})$, where $R({}^{w}\mathbf{X})$ does not intersect the object in 3D space, do not fall into any of the polygons from $Poly^{F_k}$, for $F_k \in F^{\mathcal{O}}$.

The second method proposed in this thesis is for generating synthetic datasets and an automated annotation procedure to accompany it. The foundation of this synthetic data generation pipeline is the Blender 3D suite [32]. Blender, an open-source software, offers numerous features and is not limited to any specific domain, unlike certain driving or robotics simulators. Primarily designed for computer graphics, Blender provides a wide range of tools for manipulating visual scenes. These tools include 3D object modeling, lighting and camera configurations, geometry modifications, textures and shading, image post-processing, and more. Notably, Blender can generate photorealistic renders and incorporates physics simulation using the Bullet physics engine. Moreover, Blender integrates well with DL training workflows due to its Python API, as most DL frameworks support the Python programming language. I developed a new Python-based addon called Blender Annotation Tool (BAT) to streamline the generation of segmentation mask-type annotations for synthetic visual scenes.

The efficacy of the synthetic dataset-based approach has been showcased through three experiments: a real-life robotic pick-and-place task, a benchmark (OpenLORIS-Object [7]) evaluating continual learning methods, and a grasp detection task. The results from the pick-and-place experiments highlight the enhanced performance of DL models for visual object detection achieved through the proposed synthetic data generation method. The experiments on the OpenLORIS-Object benchmark demonstrate that synthetic data can significantly increase the forward transfer of continual learning methods that utilize experience

replay. Furthermore, the grasp detection experiments showcase that due to the high flexibility and low resource requirements of the synthetic dataset generation procedure, improvements can also be achieved in grasp detection utilizing task-specific information.

To address the challenge of overcoming the "reality gap" when utilizing synthetic data, I introduced an approach that combines the proposed automated real data annotation method with the synthetic dataset generation pipeline. This approach, known as Filling the Reality Gap (FTRG), leverages a fully synthetic representation of the real-world scene and takes advantage of the ability to obtain segmentation masks for both real-world and synthetic scenes using the proposed methods. By seamlessly blending real-world and synthetic elements within a single image, this technique aims to bridge the gap between synthetic and real data. Experimental results on the pick-and-place problem demonstrate that employing the FTRG method to obtain a training dataset yields superior performance compared to state-of-the-art approaches such as using photorealistic synthetic data or domain randomization.

Forward transfer is an accuracy measure for evaluating continual learning models [7]. It measures how well a model adapts to new tasks after training on the previous ones. By utilizing the automated synthetic dataset generation pipeline, I generated a synthetic counterpart (called SynLORIS dataset) of a subset of the OpenLORIS-Object dataset and demonstrated that continual learning methods using experience replay [33] can yield superior forward transfer performance when synthetic samples were also used during their training.

The proposed synthetic dataset generation pipeline was also employed to develop a procedure for automatically generating task-specific grasp planning datasets for robotic manipulation. This method enables the fine-tuning of Grasp Quality Convolutional Neural Networks (GQCNNs) [34] for specific assembly tasks by utilizing known object and scene geometries, as well as the assembly order. The objective is to train the GQCNN models to predict feasible grasps that not only account for the object and gripper geometries and physical properties but also consider the feasibility of the grasp in relation to the subsequent placement of objects during the assembly process. The simulated experiments conducted on an asymmetric insertion-type task demonstrate that fine-tuning GQCNNs on datasets created using the proposed method significantly improves the task execution success rate.

New Scientific Results

Thesis 3

I defined and realized two procedures to create and label object segmentation datasets automatically. I showed that these datasets can be utilized to train deep learning models for visual perception tasks in robotic manipulation, such as scene recognition or object and grasp detection. The first method employs the projection algorithm (3.) to generate instance segmentation masks for real-world images with known geometry. The second method utilizes computer graphics to generate and label synthetic rendered images automatically.

Algorithm 3: Projection algorithm

input : *Image shape:* $[w, h, 3]$, *List of objects:* $\mathbf{O} = [\mathcal{O}_1, \mathcal{O}_2, \dots]$

```
/* Init annotation as black image                                    */
Init: M = zeros((w, h, 5));
for O ∈ O do
    for T ∈ O.triangles do
        /* Projection as in (6)                                      */
        v_1^i, v_2^i, v_3^i = Project(T.vertices);
        temp_img = zeros((w, h));
        /* Get internal pixels of the triangle                       */
        P = Where(DrawTriangle(temp_img, (v_1^i, v_2^i, v_3^i), color=1) == 1);
        for p ∈ P do
            if M[p][0 : 3] == [0, 0, 0] then
                /* It was background before                          */
                M[p][0 : 3] = O.color_id;
                M[p][3] = O.id;
                M[p][4] = T.id;
            else if M[p][0 : 3] == O.color_id then
                /* It is the same object                             */
                Pass;
            else
                T̃ = O.GetTriangle(M[p][3], M[p][4]);
                if IsOccluded(T, by = T̃) then
                    /* T̃ occludes T                                  */
                    Pass;
                else
                    M[p][0 : 3] = O.color_id;
                    M[p][3] = O.id;
                    M[p][4] = T.id;
return: M
```

Sub-thesis 3.1

I showed that incorporating synthetic samples during the training process of a continual learning model that utilizes experience replay can significantly en-

hance its forward transfer. I demonstrated the validity of this statement by creating a synthetic version of a subset of the OpenLORIS Object dataset and comparing two continual learning models: one that was trained using synthetic data and one that was not.

Sub-thesis 3.2

I introduced a solution to address the "reality gap" challenge when transferring deep learning models trained on synthetic data to the real world. This solution, named FTRG (Filling The Reality Gap), involves the integration of automated real and synthetic data annotation techniques to enable a smooth transition between synthetic and real components within a single image. Through comparative analysis of Mask R-CNN models trained on datasets utilizing different methods to overcome the "reality gap", including domain randomization and photorealistic synthetic data, I demonstrated that the FTRG method can achieve beyond the state-of-the-art performance.

Sub-thesis 3.3

I proposed a method for creating task-specific grasp detection datasets for robotic assembly tasks that consider grasp poses that do not necessarily result in collision-free placing. This method involves automated synthetic data generation, labeling, and sampling-based grasp planning techniques, leveraging known object and assembly geometries and assembly order. I demonstrated the effectiveness of the method by fine-tuning a GQCNN network on a generated dataset and showing that the fine-tuned GQCNN outperforms the original in an asymmetric insertion-type robotic assembly task.

Corresponding publications: [KA7, KA8, KA9, KA10, KA11].

PUBLICATIONS RELATED TO THE THESIS

[KA1] A. I. Károly, R. Fullér, and P. Galambos, "Unsupervised clustering for deep learning: A tutorial survey," *Acta Polytechnica Hungarica*, vol. 15, no. 8, pp. 29–53, 2018.

[KA2] A. I. Károly, J. Kuti, and P. Galambos, "Unsupervised real-time classification of cycle stages in collaborative robot applications," in *2018 IEEE 16th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*. IEEE, 2018, pp. 000 097–000 102.

[KA3] A. I. Károly, M. Takács, and P. Galambos, "OCSVM-based Evaluation Method for Generative Neural Networks," in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–6.

[KA4] A. I. Károly, P. Galambos, J. Kuti, and I. J. Rudas, "Deep learning in robotics: Survey on model structures and training strategies," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 1, pp. 266–279, 2020.

[KA5] A. I. Károly, R. N. Elek, T. Haidegger, K. Széll, and P. Galambos, "Optical flow-based segmentation of moving objects for mobile robot navigation using pre-trained deep learning models," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*. IEEE, 2019, pp. 3080–3086.

[KA6] A. I. Károly, R. N. Elek, T. Haidegger, and P. Galambos, "Moving Obstacle Segmentation with an Optical Flow-based DNN: an Implementation Case Study," in *2021 IEEE 25th International Conference on Intelligent Engineering Systems (INES)*. IEEE, 2021, pp. 000 189–000 194.

[KA7] A. I. Károly and P. Galambos, "Automated Dataset Generation with Blender for Deep Learning-based Object Segmentation," in *2022 IEEE 20th Jubilee World Symposium on Applied Machine Intelligence and Informatics (SAMI)*. IEEE, 2022, pp. 000 329–000 334.

[KA8] A. I. Károly, Á. Károly, and P. Galambos, "Automatic Generation and Annotation of Object Segmentation Datasets Using Robotic Arm," in *2022 IEEE 10th Jubilee International Conference on Computational Cybernetics and Cyber-Medical Systems (ICCC)*. IEEE, 2022, pp. 000 063–000 068.

[KA9] A. I. Károly, S. Tirczka, T. Piricz, and P. Galambos, "Robotic Manipulation of Pathological Slides Powered by Deep Learning and Classi-

cal Image Processing," in *2022 IEEE 22nd International Symposium on Computational Intelligence and Informatics and 8th IEEE International Conference on Recent Achievements in Mechatronics, Automation, Computer Science and Robotics (CINTI-MACRo)*. IEEE, 2022, pp. 000 387–000 392.

[KA10] A. I. Károly and P. Galambos, "Task-Specific Grasp Planning for Robotic Assembly by Fine-Tuning GQCNNs on Automatically Generated Synthetic Data," *Applied Sciences*, vol. 13, no. 1, p. 525, 2023.

[KA11] A. I. Károly, S. Tirczka, H. Gao, I. J. Rudas, and P. Galambos, "Increasing the Robustness of Deep Learning Models for Object Segmentation: A Framework for Blending Automatically Annotated Real and Synthetic Data," *IEEE Transactions on Cybernetics*, 2023.

REFERENCES

[1] H. A. Pierson and M. S. Gashler, "Deep Learning in Robotics: A Review of Recent Research," *CoRR*, vol. abs/1707.07217, 2017. [Online]. Available: http://arxiv.org/abs/1707.07217

[2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[3] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[4] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[5] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, "A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 724–732.

[6] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[7] Q. She, F. Feng, X. Hao, Q. Yang, C. Lan, V. Lomonaco, X. Shi, Z. Wang, Y. Guo, Y. Zhang *et al.*, "OpenLORIS-Object: A robotic vision dataset and

benchmark for lifelong deep learning," in *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2020, pp. 4767–4773.

[8] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, *Object Recognition with Gradient-Based Learning*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 319–345. [Online]. Available: https://doi.org/10.1007/3-540-46805-6_19

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[10] Y. Bengio, "Deep learning of representations for unsupervised and transfer learning," in *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, 2012, pp. 17–36.

[11] M. Huh, P. Agrawal, and A. A. Efros, "What makes ImageNet good for transfer learning?" *CoRR*, vol. abs/1608.08614, 2016. [Online]. Available: http://arxiv.org/abs/1608.08614

[12] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[13] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *The International journal of robotics research*, vol. 37, no. 4-5, pp. 421–436, 2018.

[14] P. Baldi, "Autoencoders, unsupervised learning, and deep architectures," in *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, 2012, pp. 37–49.

[15] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," *CoRR*, vol. abs/1511.06434, 2015. [Online]. Available: http://arxiv.org/abs/1511.06434

[16] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Boochoon, and S. Birchfield, "Training deep networks with synthetic data: Bridging the reality gap by domain randomization," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 969–977.

[17] A. Prakash, S. Boochoon, M. Brophy, D. Acuna, E. Cameracci, G. State, O. Shapira, and S. Birchfield, "Structured domain randomization: Bridging the reality gap by context-aware synthetic data," in *2019 International*

Conference on Robotics and Automation (ICRA). IEEE, 2019, pp. 7249–7255.

[18] M. Roberts, J. Ramapuram, A. Ranjan, A. Kumar, M. A. Bautista, N. Paczan, R. Webb, and J. M. Susskind, "Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 912–10 922.

[19] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[20] B. Schölkopf, R. C. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt, "Support vector method for novelty detection," *Advances in neural information processing systems*, vol. 12, 1999.

[21] D. M. Tax and R. P. Duin, "Support vector data description," *Machine learning*, vol. 54, no. 1, pp. 45–66, 2004.

[22] Z.-Q. Zeng, H.-B. Yu, H.-R. Xu, Y.-Q. Xie, and J. Gao, "Fast training support vector machines using parallel sequential minimal optimization," in *2008 3rd international conference on intelligent system and knowledge engineering*, vol. 1. IEEE, 2008, pp. 997–1001.

[23] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM transactions on intelligent systems and technology (TIST)*, vol. 2, no. 3, pp. 1–27, 2011.

[24] J. Kao, D. Tian, H. Mansour, A. Vetro, and A. Ortega, "Moving object segmentation using depth and optical flow in car driving sequences," in *2016 IEEE International Conference on Image Processing (ICIP)*, Sep. 2016, pp. 11–15.

[25] T. Zhou, S. Wang, Y. Zhou, Y. Yao, J. Li, and L. Shao, "Motion-attentive transition for zero-shot video object segmentation," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 07, 2020, pp. 13 066–13 073.

[26] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *Computing Research Repository (CoRR)*, vol. abs/1505.04597, 2015, visited on 2019-04-15. [Online]. Available: http://arxiv.org/abs/1505.04597

[27] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE*

conference on computer vision and pattern recognition, 2016, pp. 2818–2826.

[28] A. Faktor and M. Irani, "Video segmentation by non-local consensus voting." in *BMVC*, vol. 2, no. 7, 2014, p. 8.

[29] T. Sørensen, "A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on danish commons," *Biol. Skr.*, vol. 5, pp. 1–34, 1948.

[30] L. R. Dice, "Measures of the amount of ecologic association between species," *Ecology*, vol. 26, no. 3, pp. 297–302, 1945.

[31] K. M. Dawson-Howe and D. Vernon, "Simple pinhole camera calibration," *International Journal of Imaging Systems and Technology*, vol. 5, no. 1, pp. 1–6, 1994.

[32] B. O. Community, *Blender - a 3D modelling and rendering package*, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. [Online]. Available: http://www.blender.org

[33] D. Rolnick, A. Ahuja, J. Schwarz, T. Lillicrap, and G. Wayne, "Experience replay for continual learning," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/fa7cdfad1a5aaf8370ebeda47a1ff1c3-Paper.pdf

[34] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," *arXiv preprint arXiv:1703.09312*, 2017.