DELFT UNIVERSITY OF TECHNOLOGY

MASTER THESIS

Zero-shot learning in pick-and-place tasks using neuro-symbolic concept learning

Author:
N.M.VAN DER SAR

Supervisors:
C. Hernandez Corbato
F. ter Haar

A thesis submitted in fulfillment of the requirements for the degree of Master of Science

in

Biorobotics

Biomechanical Engineering

June 14, 2021

Abstract

Pick and place systems that operate in a warehouse setting have been studied a lot recently due to the high economic value for e-commerce companies. In this thesis, the focus is on the perception pipeline that performs object recognition given a certain input data stream (typically RGB-D images). Impressive results regarding object recognition have been reported in the last years, mainly driven by the development of convolutional neural networks. However, only very few proposed perception pipelines are suitable to adapt quickly to recognize new objects. This is considered a problem since large e-commerce companies add many new products to their inventory every day.

In this thesis, efforts are made to solve this problem by proposing the use of a neuro-symbolic model in which concept learning is combined with symbolic reasoning. First, visual attributes are obtained from an input image by passing it through the neural part of the model, which consists of two β-Variational Autoencoders. The key element of the model is that the visual attributes can be recognized even if a particular combination of visual attributes have not been in the training dataset. Next, given a knowledge base and the visual attributes as inferred from the input image, a symbolic reasoner infers the most likely object ID. Hereby, the knowledge base is manually constructed and describes the relationships between object IDs and the corresponding visual attributes. The implementation of the neuro-symbolic model is first tested on a synthetic dataset which is similar to the dataset as used in the study that the neural part is based upon. Thereafter, the neuro-symbolic model is tested on real RGB-D images of the pick-and-place dataset and several iterations of the baseline model are evaluated. Hereby, the main research question is formulated as: *Can a neuro-symbolic model be used to recognize unseen objects given RGB-D data as typically seen in pick and place scenarios?*

The best top-1 accuracy score on unseen images of the synthetic dataset was 79.5%. However, using the same neuro-symbolic model on the pick-and-place dataset, the top-1 accuracy score on unseen images was only 25.5%. In the following iterations of the model, the top-1 accuracy score was improved up to 32.4%. Analysis of the results of the pick-and-place experiments shows that the neural part is not very capable of recognizing the correct visual attributes. This likely be due (to some extend) to the simulation-to-real gap. However, further research is required to identify the exact cause(s) of the performance drop. Concludingly, the proposed neuro-symbolic model is capable of recognizing unseen images of the synthetic dataset, but is not very capable of recognizing unseen images of the pick-and-place dataset.

“Learning never exhausts the mind.”

Leonardo da Vinci

Contents

1. Introduction

1.1 Motivation

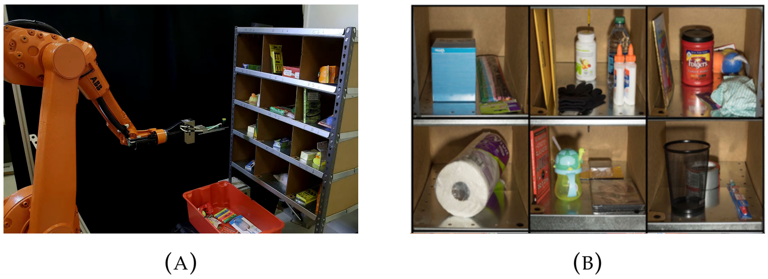
Pick-and-place systems

Picking and placing of objects is an elementary robotic skill that is required in many applications. Some examples are 1) a rover on another planet has to pick up rocks and place it in a machine for analysis, 2) a service robot that has to pick up bread spreads and place it on a table or 3) a pick-and-place system in a warehouse environment has to pick up products and place it in a tote to fulfill an order. Research related to the latter application has largely been pushed by big e-commerce companies due to the high economic value for such pick-and-place systems [1, 2, 3]. It also provides a safe and controllable test environment for fundamental research. A pick-and-place system can generally be broken down into three subsystems [40]:

1. *Perception pipeline* that performs object recognition and sometimes includes a pose estimation of the object.
2. *Grasping pipeline* that plans how the object can be picked up and executes that plan.
3. *Movement pipeline* plans and executes the movement from the grasp position to the target position.

Perception pipelines in a warehouse environment

In this thesis, the focus is on the perception pipeline of a pick-and-place system in a warehouse environment. A typical scene of this environment is depicted in figure 1.1. Traditionally, feature matching methods have been used to recognize objects [28, 32, 53]. Since the renaissance of deep learning in computer vision in 2012 [45], most perception pipelines that have been proposed involve the use of convolutional neural networks (CNNs) [13, 37, 41, 52]. These methods outperform feature matching methods with respect to runtime and robustness against (partial) occlusion and difficult lighting conditions (e.g. shadows and reflections) [4, 15, 31].


(A)

(B)

FIGURE 1.1: (A) Example of a pick-and-place scenario in a warehouse setting. Adapted from [51]. (B) Example of shelves from which a specific object has to be picked up. Adapted from [13] and edited.

Limitations of current perception pipelines

The main disadvantages of convolutional neural networks are the vast amount of training data that is required in the training phase and the low adaptability to new objects [18, 40]. High adaptability of the perception pipeline is required since many new products are added to the store everyday in large e-commerce companies [3]. Most of the proposed perception pipelines require retraining of the network with additional training data of the new object [9, 38, 41]. This process is quite slow and inefficient since previously obtained knowledge is not being used in a smart manner. That is, by retraining the network a new mapping is created between the image of the new object and a corresponding new label. Thus, instead of leveraging from the initial training phase, the new objects are just being handled as if the neural network has to be trained from scratch. Although efforts have been made to increase the adaptability of the perception pipeline [52], this is yet considered an open problem [40].

1.2 Neuro-symbolic model and zero-shot learning

In this thesis, the adaptability problem will be approached as a zero-shot learning problem, in which the goal is to classify objects that are present not the training data [48, 50]. We propose a neuro-symbolic model that addresses this problem by using concept learning (i.e. neural part) in combination with symbolic reasoning and a knowledge base (i.e. symbolic part). Concept learning allows for learning visual attributes of objects instead of labels [24]. Subsequently, symbolic reasoning is applied to find the object ID given these visual attributes and a knowledge base. For example, an object ID could be Pink Lady, whereby the corresponding visual attributes are red and yellow for object colour and apple for object shape. In figure 1.2, a high-level overview of the pipeline that we propose is depicted. The following steps are taken to identify an object:

1. An input image that consists of RGB data, whether or not in combination with depth data, is fed to the neural part of the pipeline.

2. The neural part of the model processes the input image and outputs a tuple of detected visual attribute values. The values in this tuple are symbolic and by obtaining this tuple, the neural part of the model is concluded. The concept learning capabilities of the perception pipeline are fully embedded in this part of the model.

3. Subsequently, the tuple of detected visual attribute values together with a manually constructed knowledge base are inputted to a symbolic solver. These three components make up the symbolic part of the model.

4. Lastly, the symbolic solver calculates the most likely object ID based upon the inputs.

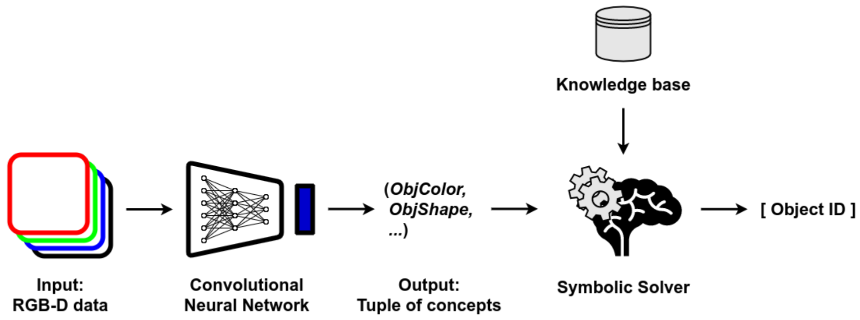


FIGURE 1.2: High level overview of the proposed neuro-symbolic model.

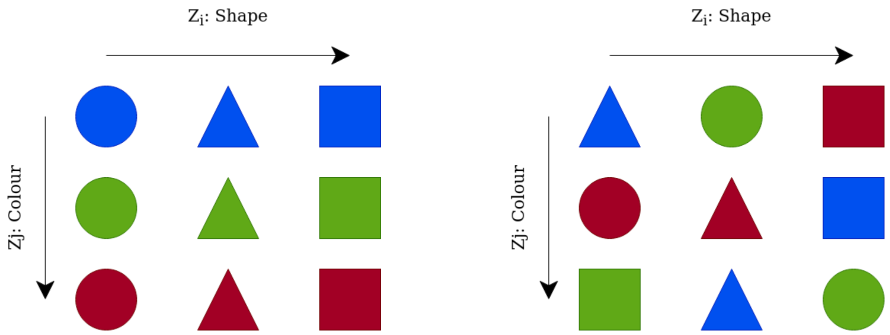
Recognizing unseen objects

The tuple of visual attribute values is compositional in nature and this property can be used to obtain the object ID of unseen objects. To do so, there are two requirements that need to be met: 1) the visual attribute values of the unseen objects need to be present in the training data and 2) there needs to be an entry in the knowledge base that maps the the tuple of visual attribute values (which thus has an unseen combination) to a certain object ID. If these requirements are met, the symbolic solver is able to obtain the correct object ID for an unseen object.

An example of how unseen objects can be identified: Consider the pipeline has successfully been trained on images of tomatoes and bananas after which the neural part of the pipeline is able to recognize 2 object shapes and 2 object colours, being 'tomato shape', 'banana shape', 'red' and 'yellow' respectively. After the correct mappings are made in the knowledge base (e.g. 'tomato' = {'tomato shape, 'red'}), the perception pipeline is able to recognize tomatoes and bananas by inputting the knowledge base together with the obtained tuple from the neural part to the symbolic solver. However, the added value of using a concept learning framework is that combinations of visual attribute values that have not been in the training data can also be recognized. In this case, the unseen tuples would be {'tomato shape', 'yellow'} and {'banana shape', 'red'}. In the knowledge base, a mapping can be created from these tuples to a particular object ID (e.g. 'tomana' = {'tomato shape', 'yellow'} and 'banato' = {'banana shape', 'red'}). If the entry exists in the knowledge base, the symbolic solver will be able to find the object ID if such an unseen yellow tomato or red banana is depicted in the input image.

Neural part of the model

The neural part of the model that we propose is based on the Symbol-Concept Association Network (SCAN) as proposed by Higgins et al. [24]. One of the strengths of SCAN is that it is capable of learning a latent space in an unsupervised manner in which distinct visual attributes are encoded independently [24]. In the literature, this property is called *disentanglement* (see Figure 1.3) [7] and has recently been studied extensively because a disentangled representation enables abstract visual reasoning, increases the explainability of the model and decreases the need for labeled training data [11, 22, 35, 44]. A β-Variational Autoencoder (β-VAE) [22] is used to create such a disentangled representation of the input image, which will be called β-VAE$_d$. This is an unsupervised method, hence after training only a sub-symbolic (i.e. numeric) latent vector for that particular image can be obtained and not the visual attribute values that are needed for the symbolic reasoning part of the model. Another β-VAE, named β-VAE$_{sym}$, is trained to create a mapping between the (symbolic) visual attribute values and the corresponding (numeric) latent vectors. This process is called *grounding* and this is one technique that solves the symbolic grounding problem [21].



(A) Disentangled latent space.

(B) Entangled latent space.

FIGURE 1.3: Schematic representation of a disentangled and an entangled latent space. Consider you have a blue filled circle as a reference image and the latent vector has been calculated for that image (e.g. by using the encoder of an autoencoder). Then the latent space is considered disentangled when there is one latent factor encoding one visual attribute. In this example, the two main visual attributes of the blue filled circle are the shape and the colour. As can be seen in subfigure A, one latent factor encodes for shape (i.e. z_i) and one latent factor encodes for colour (i.e. z_j). Thus, this is considered a disentangled latent space. For subfigure B, this is not the case: there is no latent factor clearly encoding either shape or colour. In other words, the latent space is entangled.

Symbolic part of the model

After the β-VAE$_{sym}$ has been trained, the tuple of visual attribute values can finally be obtained. In symbolic reasoning part of the model, the tuple of visual attribute values as inferred by the neural network are compared with the visual attribute values from the provided knowledge base that describe the unseen objects. This is done by mapping the inferred attribute values and the knowledge base attributes values to a sub-symbolic (i.e. involving numeric values) helper space after which a nearest neighbor calculation is applied.

1.3 Research Questions

The main research question is defined as:

Can a neuro-symbolic model be used to recognize unseen objects given RGB-D data as typically seen in pick-and-place scenarios?

To answer the main research question, several subquestions are formulated:

1. *Is the neuro-symbolic model capable of recognizing unseen images of the Deepmind Lab dataset?*
   The motivation behind this question is twofold:

   (a) There are significant differences between the Deepmind Lab dataset which was used in the original study of SCAN [24] and the Pick-and-Place dataset which is used in this study. That is, the Deepmind Lab dataset consists of synthetic images in which every image depicts a unique combination of visual attributes. On the other hand, the Pick-and-Place dataset used here shows real images, whereby many images depict the same object but from a different camera angle. By using the original Deepmind Lab dataset [6], the hypothesis that the proposed neuro-symbolic model is able to recognize unseen images can be tested without a potential performance decrease caused by the used dataset (e.g. due to the reality gap).

(b) By conducting an experiment using similar data as the Deepmind Lab dataset, it is possible reproduce some of the results as presented by Higgins et al. [24]. This allows for validating their results and additionally, it enables validating the implementation of SCAN as used in this study since the original code base is not publicly available.

2. *What is the influence of disentanglement on the top-1 accuracy of recognizing unseen images?*
   In the literature, it has been stated that high disentanglement of the latent space is important for visual reasoning [18, 24]. This subquestion aims at testing that statement.

3. *Does depth information increase the top-1 accuracy of recognizing unseen objects?*
   In pick-and-place scenarios, RGB-D cameras are typically used. The depth information provides extra information about the object and thereby might lead to higher top-1 accuracy performance on unseen objects.

4. *How does the capacity of the neural network(s) influence the performance of the neuro-symbolic model?*
   An intuitive explanation of the capacity of a neural network is how much information a neural network is able to encode. Neural networks with many hidden layers generally have a larger capacity. Several studies have described the importance of the capacity of a neural network [5, 46]. This subquestion examines if that is also the case for the performance of the neuro-symbolic model.

5. *Can the loss function as used in SCAN be modified to achieve a better performance of the neuro-symbolic model?*
   The loss function dictates the structure of the latent space. Thus by altering the loss function, the performance of the neuro-symbolic model may be enhanced.

1.4 Contributions

The main contributions of this thesis are threefold:

- *Propose a neuro-symbolic model to recognize unseen objects in a pick-and-place scenario as typically seen in a warehouse setting.* This is a novel approach to this scenario and is aimed at increasing the adaptability compared to most other perception pipelines that have been proposed in this scenario.

- *Extend SCAN [24] with a symbolic solver.* In the work of Higgins et al. [24], the focus was mainly on testing the capabilities of SCAN in a synthetic game environment. Hereby, only a brief example was provided about the possibilities regarding unseen images. Furthermore, there was no symbolic solver proposed to obtain an object ID since SCAN was not originally proposed to identify unseen objects, but to decompose an input image into visual attributes. This study extends the work of Higgins et al. [24] by proposing a symbolic solver that serves the purpose of obtaining an object ID and by proposing and testing some iterations of the neural network architecture.

- *Apply SCAN to a different domain.* The domain of interest in this study is a pick-and-place scenario in a warehouse setting whereby real images have been used, which contributes to define the applicability of SCAN.

1.5 Thesis Outline

In **chapter 2**, a brief overview of related work will be provided. Next, in **chapter 3**, the details about the neuro-symbolic model as proposed here are described. Thereafter, in **chapter 4**, the experiments are explained and the results are presented. Hereby, two sets of experiments are distinguished: Deepmind Lab experiment (section 4.1) and the Pick-and-Place experiments (section 4.2). In the Deepmind Lab experiment, the proposed model is applied to reproduce a part of the results as presented by Higgins et al. [24] and test if the proposed model could work to recognize unseen images. Subsequently, in the Pick-and-Place experiments, the model is tested for a pick-and-place scenario in a warehouse setting. In both these experiments, there is a test dataset consisting of unseen images and a symbolic description of these unseen images. In **chapter 5**, conclusions will be drawn based upon the results of the experiments. Here, the research question and corresponding subquestions are answered and leads for future work are proposed.

2. Related Work

2.1 SCAN

In the work of Higgins et al. [24], a neural network framework called Symbol-Concept Association Network (SCAN) is proposed for learning visual concepts in a game environment. By using a neural network that involves multiple β-VAEs [22], Higgins et al. [24] were able to learn a disentangled latent space in a largely unsupervised manner, whereby only a few labeled images were required to solve the symbolic grounding problem [21]. The original paper mainly focuses on how well concepts could be learned, whereby most experiments were conducted using images generated from the Deepmind Lab game environment [6]. However, it has been stated in the literature that a disentangled latent space could potentially be applied to create a neuro-symbolic model that allows for complex visual reasoning [18]. In this thesis, that statement is tested by extending SCAN with a symbolic part and applying the neuro-symbolic model to recognize unseen items in a pick-and-place scenario in a warehouse setting.

2.2 Pick-and-Place pipelines

As stated in the introduction, most perception pipelines proposed in the literature require retraining the perception pipeline to adapt to new objects [40]. The perception pipeline as proposed by Zeng et al. [52], however, does not require retraining and can be adapted fast to recognize unseen objects. Hereby, Zeng et al. [52] proposed the use of metric learning to be able to recognize unseen objects in a pick-and-place scenario. In metric learning, images are mapped to a feature space whereby neighbouring points have similar properties. During training, this mapping is constructed based upon the training data. Merely a feature vector describing the properties of the image is outputted by the network (i.e. no label) and thus also the feature vector of images that were not in the training dataset can be computed [25]. Since images of similar properties are mapped to the same region in feature space, new objects can be identified by comparing the distance (i.e. typically L2 distance) between a labeled, but unseen reference image and the input image of the scene [52]. In the paper of Zeng et al. [52], the best score reported on the top-1 accuracy was 82.1% for unseen objects. The results from this thesis will be compared against that score since this is state-of-the-art for the considered problem and domain. Although the results are quite impressive, the perception pipeline as proposed by Zeng et al. [52] does suffer from the disadvantage that the learned metric space is entangled and thereby hard to interpret by humans. Additionally, due to the entanglement, this perception pipeline does not allow for symbolic reasoning in its current form.

2.3 Zero-shot learning

The model proposed by Zeng et al. [52] focuses on the same application domain as this study, namely a pick-and-place task in a warehouse setting. However, in its abstraction, the addressed problem of recognizing unseen objects have long been studied albeit in other domains. It is known as the zero-shot learning problem [50]. Xian et al. [50] have reviewed and benchmarked many different zero-shot learning frameworks on multiple different datasets. Hereby, the reported top-1 accuracy ranged between 51.3% and 80.6%. It must be stated that the problem addressed here is different from the one analyzed by Xian et al. [50]. However, it provides some context for the best scores reported in this thesis. It is beyond the scope of this thesis to discuss all the different types of zero-shot learning frameworks. Interested readers are recommended to read the review paper of Xian et al. [50]. Since the neuro-symbolic model as presented in this thesis falls under a subcategory called Compositional zero-shot learning, this particular branch will be discussed next.

2.3.1 Compositional zero-shot learning

Compositional zero-shot learning (CZSL) is a particular subcategory of zero-shot learning that focuses on recognize unseen compositions of visual attribute values [39]. The study of Lampert et al. [33] is one of the first models that describes this approach. This is done by training a classifier to recognize visual attributes either directly (Direct Attribute Approach) or indirectly (Indirect Attribute Prediction). Jayaraman et al. [27] iterate on the work of Lampert et al. [33] and propose the use of a random forest approach to account for the unreliability of attribute prediction. These approaches More recently, Graph Convolutional Networks (GCNs) have been used in CZSL in which graphs are used to embed information about the image like visual attributes and object relations [30, 39, 49]. One of the main benefits of using graphs and GCNs is that it is well able to implement dependencies within data [39].

The model presented in this thesis is trained in a largely unsupervised manner due to using SCAN [24], which is in contrast to the studies mentioned above that rely on a significant amount of attribute-labeled and class-labeled data. Further, typically, zero-shot learning studies use a dataset that is constructed for evaluating a certain model consisting of visual attribute rich images (e.g. [12, 26, 47]). Although such datasets could work well for benchmarking a certain model and comparing it with other models, it doesn't show the applicability to real-life applications where less variety of images may exists. In contrast, in this thesis, the application (i.e. a pick-and-place scenario in a warehouse setting) is the main focus point.

3. Methods

In this chapter, the neuro-symbolic model that we propose will be explained in full detail. First, in section 3.1, the neural part of the model will be discussed. The original design of the neural network is called SCAN [24] and involves two β-Variational Autoencoders (β-VAE) and one Denoising Autoencoder (DAE). In this work, SCAN is tested in its original form and also several iterations that we propose are tested. Next, in section 3.2, the symbolic part of the model is discussed. This part extends the neural part and enables obtaining the object ID of an unseen object given a knowledge base and the tuple of visual attribute values as outputted by the neural part of the model. Hereby, the symbolic part is proposed by the author of this thesis. In figure 3.1, the proposed neuro-symbolic model is depicted including the most important details. The steps for identifying an object are:

1. The input image is fed to the β-VAE$_d$.
2. The latent representation of the input image from the β-VAE$_d$ is passed as an input to the decoder of the β-VAE$_{sym}$.
3. The β-VAE$_{sym}$ maps the latent representation to a k-hot vector, which can be rewritten to a tuple with a symbolic description of the inferred visual attributes.
4. The tuple of inferred visual attribute values and the description of the unseen objects from the knowledge base are mapped to the helper space. This helper space allows for similarity comparison of visual attributes as inferred and from the knowledge base (e.g. object colour and object shape).
5. A nearest neighbor lookup is executed in the helper space to obtain the object ID of the unseen object.

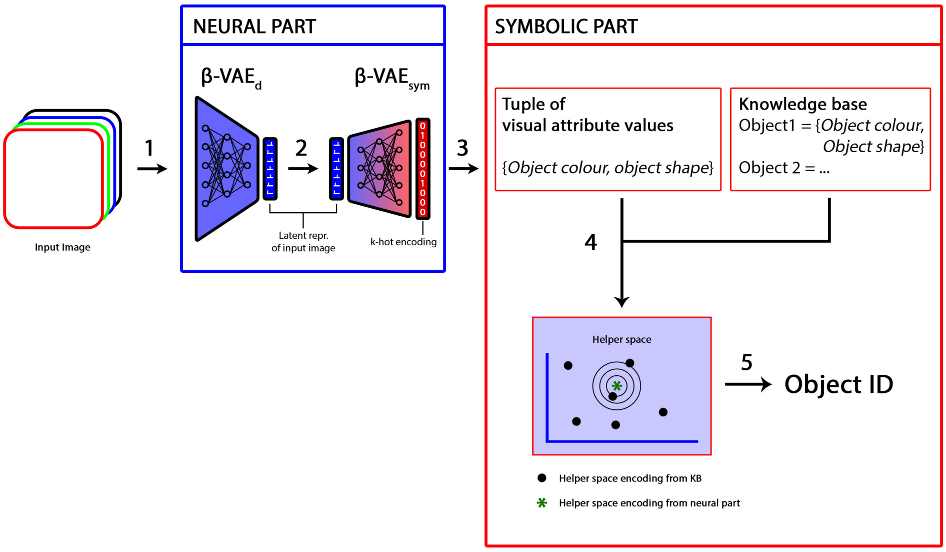

NEURAL PART
β-VAE_d
β-VAE_sym
Input Image
1
2
3
Latent repr. of input image
k-hot encoding
SYMBOLIC PART
Tuple of visual attribute values
{Object colour, object shape}
Knowledge base
Object1 = {Object colour, Object shape}
Object 2 = ...
4
Helper space
5
Object ID
● Helper space encoding from KB
✱ Helper space encoding from neural part

FIGURE 3.1: The proposed neuro-symbolic model.

3.1 Neural Part

In general, autoencoders consist of two neural networks: an encoder and a decoder. The encoder network compresses the input image to a smaller space which is called the latent space. Subsequently, the decoder network takes some latent representation vector and decompresses it to reconstruct the input image. To train this architecture, the loss function consists of a term that relates to the difference between the reconstructed and original input image (i.e. the reconstruction loss). However, the loss function can be extended to achieve certain goals in the behaviour of the autoencoder, for example creating a disentangled latent space. In SCAN [24], β-VAE's are used to create a disentangled latent representation from the input image (in case of β-VAE$_d$) and to obtain the symbolic attribute values from that disentangled latent representation (in case of β-VAE$_{sym}$). We adopted this approach and made several modifications of the neural network architecture and loss function which will be discussed in the following subsections.

3.1.1 β-VAE$_d$

Loss function

The loss function of a typical β-VAE is as follows [22, 29]:

$$\mathcal{L}_x(\theta, \phi, \mathbf{x}, \mathbf{z_x}, \beta) = \mathbb{E}_{q_\theta(\mathbf{z_x}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z_x})] - \beta\, D_{KL}(q_\phi(\mathbf{z_x}|\mathbf{x}) \mid\mid p(\mathbf{z_x})) \tag{3.1}$$

where θ and ϕ - distribution parameters of the encoder and decoder respectively; p and q represent the prior and posterior distributions; $\mathbf{x}$ - input image; $\mathbf{z_x}$ - latent space representation of input image $\mathbf{x}$; β - disentanglement parameter where larger β puts more emphasis on disentanglement; $\mathbb{E}_{q_\theta(\mathbf{z_x}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z_x})]$ - the expected reconstruction error also known as cross entropy; D_{KL} - Kullback-Leibler (KL) divergence term that indicates the similarity between the latent space distribution of the input image and the normal Gaussian prior distribution.

This loss function defines the reconstruction error in pixel space. That causes a problem for small areas in the image that contain a lot of information since the relative contribution to the loss will be small compared to the larger areas. Alternatively, Higgins et al. [22] propose to replace the log likelihood term by a L2 loss term in a latent space. To be able to do that, a DAE was trained and the latent space thereof was used in the new lost function:

$$\mathcal{L}_x(\theta, \phi, \mathbf{x}, \mathbf{z_x}, \beta) = ||J(\hat{\mathbf{x}}) - J(\mathbf{x})||^2 - \beta\, D_{KL}(q_\theta(\mathbf{z_x}|\mathbf{x}) \mid\mid p(\mathbf{z_x})) \tag{3.2}$$

where $J(\mathbf{x})$ - function that maps input image $\mathbf{x}$ to the latent space of the DAE; $\hat{\mathbf{x}}$ - reconstructed image given input image $\mathbf{x}$. In this study, the method of using the reconstruction loss in the DAE latent space is applied unless stated otherwise.

Original architecture

The architecture of the β-VAE$_d$ as proposed by Higgins et al. [24] is described as follows: The input image has a dimension of 80x80x3. This image is then passed to the encoder part of the β-VAE$_d$ (step 1 of figure 3.1) that consists of 4 convolutional layers, which have 32, 32, 64 and 64 filters respectively. The filter size is 4x4 and the stride length is 2 in horizontal and vertical direction. After the convolutional layers, there is one fully connected layer with a size of 256 neurons. Subsequently, the fully connected layer is connected to a 64 dimensional latent space that encodes 32 independent Gaussian distributions as the β-VAE$_d$ is a variational autoencoder [29]. The decoder part of the β-VAE$_d$ has the same properties as the encoder, but uses deconvolutional layers instead. The activation functions and padding algorithm used are ReLU's and SAME respectively. To train the neural network, the ADAM optimization algorithm is used with the learning rate set to 1e-4 and $\epsilon = 1e - 8$.

In order to train the β-VAE$_d$, a DAE is required as explained above. The architecture of this neural network is largely the same as the architecture of the β-VAE$_d$. The only difference is that it has a latent space of 100 neurons (which don't encode any Gaussian distribution since the DAE is not a type of variational autoencoder).

Assessing the level of disentanglement

Efforts have been made to quantify the level of disentanglement. However, there is yet no consensus which metric is suitable to quantify disentanglement [10, 42]. In this thesis, a highly disentangled latent space is defined to have the following properties:

1. *A visual attribute is encoded by maximally one latent factor.* This property will be referred to as the *independence* property as the visual attribute only depends on one latent factor.

2. *A latent factor encodes maximally one visual attribute.* This property will be referred to as the *uniqueness* property as the latent factor only encodes one visual attribute.

To determine the level of disentanglement of the latent space, visual analysis is applied and additionally the matrix of informativeness [14] is constructed and examined.

Visual analysis

For visual analysis the images are constructed as follows: First, the latent vector encoding a reference image is calculated. Next, the reference image is reconstructed, but a particular latent factor is varied (for example, see figure 4.4). However, the visual analysis is only based on one reference image from the training dataset, thus not representative for the whole dataset. On the other hand, the matrix of informativeness is constructed based upon all training images and therefor used as support for the visual analysis.

Matrix of informativeness

The informativeness matrix [14] is used since it's intuitive and allows for inspection of the latent space considering all training images. In figure 3.2, an example is shown whereby on the x-axis the four visual attributes are represented and the y-axis consists of the 32 elements of the latent vector. The i, j-th value of the matrix is equal to the mutual information between the j^{th} visual attribute and the i^{th} latent factor. This measure originates from information theory and is mathematically described as:

$$I(c_j; z_i) = H(z_i) - H(z_i|c_j) \tag{3.3}$$

Where $H(z_i)$ is describes the entropy of latent factor z_i. Formally, the mutual information can be described as how much information a certain random variable communicates about another random variable [19]. In this thesis, the *mutual information* is calculated by comparing the change of the latent factors (which is calculated)

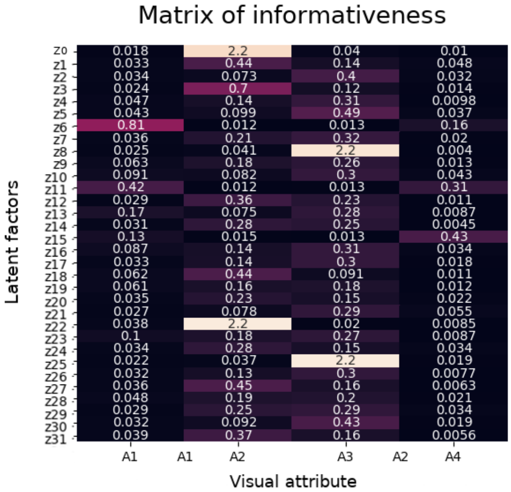


FIGURE 3.2: An example of matrix of informativeness from Deepmind Lab experiment (see section 4.1).

with the change of the ground truth attribute labels (that are given) for a particular set of images. If the i, j-th value of the matrix is high, it means that the mutual information is high. And thus that when the j-th element from the latent vector changes the i-th visual attribute value changes. As an example: in figure 3.2, there is a (relatively) high score for latent factor C23 and visual attribute A2, which could be wall colour for example. Then, from this matrix, it is expected that by varying latent factor C23, a change of visual attribute A2 (wall colour) would be observable.

Properties of results for a highly disentangled latent space
In case the disentanglement properties as discussed above are met, the following properties would be observable in the results of the visual analysis and matrix of informativeness:

1. *Independency property:* In the graphical representation, one particular visual attribute only changes for one of the latent factors (i.e. for only one row of the image, a certain visual attribute changes). With respect to the matrix of informativeness, it means means that for one particular visual attribute, there is maximally one latent factor for which the mutual information is high.
2. *Uniqueness property:* Graphically, this means that by changing a certain latent factor, maximally one visual attribute is changed (e.g. object colour). Regarding the matrix of informativeness, it means that a latent factor has a high mutual information score with maximally one visual attribute.

If these properties apply, the matrix of informativeness would look similar to the dummy example as shown in figure 3.3. Note that, the independency and uniqueness properties are related but different. That is, if the independency property is not met, there would be two lighter cells (i.e. high mutual information score) in a particular column. On the other hand, if the uniqueness property is not met, there would be two lighter cells (i.e. high mutual information score) in a particular row.

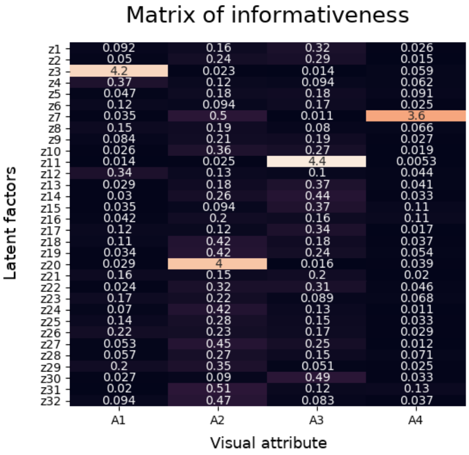


FIGURE 3.3: Example of a matrix of informativeness of a highly disentangled latent space. Dummy values were used to create this matrix.

3.1.2 β-VAE$_{sym}$

Loss function

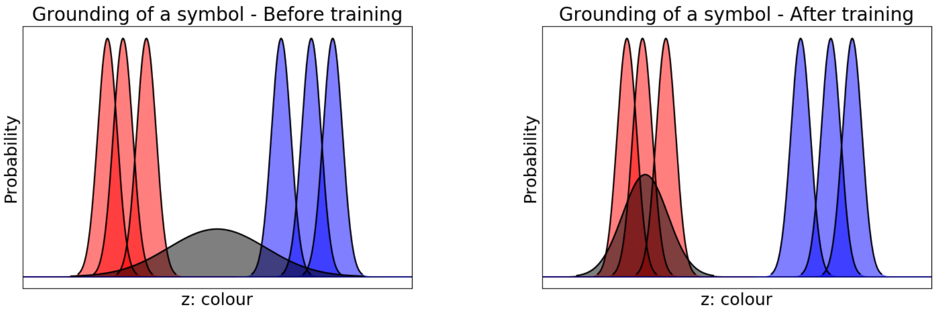
The symbolic grounding step is done by minimizing the KL divergence term between the concept distributions and the visual primitive distributions:
$D_{KL}(q(\mathbf{z_y})\,||\,q(\mathbf{z_x}))$. The latent factors relevant to the concept will have narrow distributions, whereas irrelevant latent factors will have distributions close to the prior (i.e. a normal Gaussian distribution). Intuitively, this makes sense: consider the concept (red, cube, small). In that case, only the latent factors that encode for object colour, object shape and object size are relevant and specifically defined. This will result in a narrower distribution. However, other latent factors may encode for lighting angle, object orientation or object translation and possess large variety without violating the concept (red, cube, small). The loss function of β-VAE$_{sym}$ is adopted from Higgins et al. [24] and is defined as:

$$\begin{aligned}\mathcal{L}_y(\theta_y, \phi_y, \mathbf{y}, \mathbf{x}, \mathbf{z_y}, \beta, \lambda) = \mathbb{E}_{q_{\phi_y}(\mathbf{z_y}|\mathbf{y})}[\log p_{\theta_y}(\mathbf{y}|\mathbf{z_y})] - \beta\, D_{KL}(q_{\phi_y}(\mathbf{z_y}|\mathbf{y})\,||\,p(\mathbf{z_y})) \\ - \lambda\, D_{KL}(q_{\phi_x}(\mathbf{z_x}|\mathbf{x})\,||\,q_{\phi_y}(\mathbf{z_y}|\mathbf{y}))\end{aligned} \tag{3.4}$$

where $\mathbf{y}$ - symbols that describe a concept; $\mathbf{z_y}$ - concept latent space (symbolic); $\mathbf{z_x}$ - visual primitive latent space (non-symbolic); $\mathbf{x}$ - input images that correspond to concepts $\mathbf{z_y}$. β and λ - weighting factors for disentanglement and generalisation over instances respectively. When β-VAE$_{sym}$ is trained, the latent space of the β-VAE is already disentangled in an unsupervised manner as explained in the previous part. It merely has to span the Gaussian distribution that is attached to a particular visual attribute value over the instances of input images that belong to that visual attribute value (see figure 3.4).

Architecture

The architecture of the β-VAE$_{sym}$ is almost identical to architecture as proposed by Higgins et al. [24] in any of the experiments in this study. Only minor changes have been made due to different dimensions of input images and/or number of visual attributes present in the dataset. The architecture of β-VAE$_{sym}$ is as follows: The input layer has 32 units, which resembles the dimensions of the latent space of the



(A)

(B)

FIGURE 3.4: Grounding process. The red and blue distributions represent the distributions for individual instances of that visual attribute value. That is, there are 3 red and 3 blue objects. However, the clustering of this property is done in an unsupervised manner. Therefore, it is not yet known which cluster is red and which cluster is blue. The β-VAE$_d$ has only learned to map instances with similar colours close to each other in the latent space. Subsequently, in the β-VAE$_{sym}$, the label 'red' is introduced (subfigure A) and during training it is learned to span over all the given labeled instances of the colour label 'red' (subfigure B).

β-VAE$_d$ (see step 2 in figure 3.1). Subsequently, there is a fully connected layer with 100 neurons. Lastly, another fully connected layer is implemented that has n output units. Each unit of the output layer represents a particular known visual attribute value, this is known as a k-hot encoding. Thus, n resembles the number of unique visual attribute values present in the knowledge base. The values of the k-hot vector show what visual attribute values the neural part of the model inferred given the input image. That is, a visual attribute value may be inferred to occur in the input image (element value = 1) or not (element value = 0). Thus, the k-hot vector is just another encoding for the tuple of inferred visual attribute values (see step 3 in figure 3.1). For the Deepmind Lab experiment (section 4.1), there are 16 colours for the wall colour, floor colour and object colour and 3 object shapes. This results in $n = 51$ for that experiment. Regarding the Pick-and-Place experiments, there are 9 colours and 44 shapes, resulting in $n = 53$. Again, the ADAM optimizer is used with a learning rate of $1e - 4$ and $\epsilon = 1e - 8$.

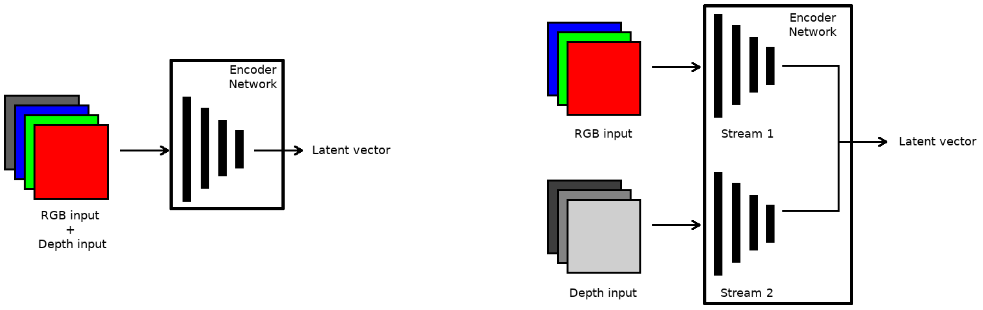
3.1.3 Iterations of the neural network architectures

The loss function and architecture described above are adopted from the work of Higgins et al. [24] and are used in the Deepmind Lab experiment and experiment 1 and experiment 3.1 of the Pick-and-Place experiments. The other experiments conducted in this study require modifications of the original design. Per experiment, these modifications are:

- *Experiments 2.1 and 2.2:* For this experiments, the input layer and output layer of the DAE, the β-VAE$_d$ and the β-VAE$_{sym}$ are extended by one layer (experiment 2.1) or three layers (experiment 2.2). This was done so that depth data or HHA data, which is a particular depth encoding, together with the RGB data could be inputted to the β-VAE$_d$. This is known as a one-stream architecture.

- *Experiments 2.3 and 2.4:* All neural networks have been adjusted to have a dedicated neural network for every data stream (i.e. RGB and depth data). In the last step of the encoder, the data is fused into a shared latent space with a size of 32. A schematic representation of a one-stream and two-stream architecture are depicted in figure 3.5.
- *Experiments 3.2 and 3.3:* In experiment 3.2, the capacity of the neural network is lowered by removing the last hidden layer of the encoder and decoder of the β-VAE$_d$. On the other hand, in experiment 3.3, an extra hidden layer is added after the input layer with 32 filters, filter size 4x4 and stride = 1.
- *Experiments 3.4:* In this experiment, the loss function as described in equation 3.2 is changed so that the reconstruction loss is not described in the latent space of the DAE, but in the pixel space of the image. This results in the following loss function for the β-VAE$_d$ in this experiment:

$$\mathcal{L}_x(\theta, \phi, \mathbf{x}, \mathbf{z_x}, \beta) = ||\hat{\mathbf{x}} - \mathbf{x}||^2 - \beta\, D_{KL}(q_\theta(\mathbf{z_x}|\mathbf{x}) \;||\; p(\mathbf{z_x})) \tag{3.5}$$



(A) One-stream encoder

(B) Two-stream encoder

FIGURE 3.5: Schematics of a one-stream and two-stream architecture. In the one-stream architecture, the RGB and depth input data are simply concatenated. The dimensions of the input image become either 80x80x4 or 80x80x6 given the concatenated normal depth image or the HHA depth image respectively. Only the input layer of the neural network is adjusted to account for the extra layer(s). In the two-stream architecture, each stream (i.e. RGB and depth input data) have their own dedicated neural network. In the last layer of the encoder network, a fully connected layer combines information from the two streams to create a single latent space.

3.1.4 Code base

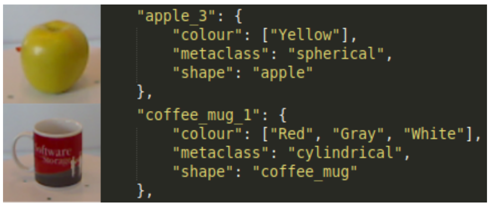
The original code from Higgins et al. [24] was not open-source, instead a recreated codebase was used as a basis for the neural part of the model in this study[1]. This code has been fully reviewed and debugged before using it in any of the experiments.

[1] https://github.com/miyosuda/scan/

3.2 Symbolic Part

3.2.1 Knowledge base

The knowledge base contains information about the visual attributes of the unseen objects and binds them to a particular object ID. In the Pick-and-Place experiments, only the visual attributes object shape and object colour are used, but it could contain other information as well (e.g. azimuth and size). The information in the knowledge base was constructed manually and saved as a JSON file (see figure 3.6).



```
"apple_3": {
    "colour": ["Yellow"],
    "metaclass": "spherical",
    "shape": "apple"
},
"coffee_mug_1": {
    "colour": ["Red", "Gray", "White"],
    "metaclass": "cylindrical",
    "shape": "coffee_mug"
},
```

FIGURE 3.6: Sample of the knowledge base as a JSON file with the symbolic description of the objects.

3.2.2 Reasoner

To obtain the most likely object ID of an unseen object, the reasoner compares the tuple of inferred visual attributes with the tuples of visual attributes from the knowledge base. Hereby, the tuples in the knowledge base are coupled to object IDs as a key-value pair, which allows for retrieving the object IDs of unseen objects. The comparison of the tuples from the neural part of the model and the knowledge base is done as follows:

1. The inferred tuple of visual attribute values and all the tuples of visual attribute values in the knowledge base are mapped to a helper space (see step 4 in figure 3.1). Hereby, two mapping functions are described in the knowledge base: one for colours and one for shapes. These mapping functions are discussed below.

2. A nearest neighbor lookup is performed for each of the visual attributes independently. Hence, if there are 4 visual attributes (e.g. in case of the Deepmind Lab experiment), there will be 4 distance values.

3. The distances are summed to a single distance value.

After comparing the inferred tuple of visual attribute values with all the tuples in the knowledge base, the minimum summed distance is selected and the object ID is obtained from the knowledge base (see step 5 in figure 3.1).

Colour

Colours are mapped to a HSV helper space. In the knowledge base, the mapping between the visual attribute value and corresponding normalized HSV value is described. For example, 'Blue' would map to: $normalize(240, 100, 100) = (0.67, 1, 1)$. The distance is defined as the Euclidean distance between the two vectors (i.e. one from the inferred tuple and one from the tuple from the knowledge base).

Shape

To compare shapes, every unique shape is mapped to an independent axis in the shape helper space. This creates a binary outcome while comparing shapes from the inferred tuple with the tuples from the knowledge base: either the distance is 0 or $\sqrt{2}$. This method is applied as such in the Deepmind Lab experiment (section 4.1). However, a slightly more fine-grained approach is implemented in the Pick-and-Place experiments by including a *meta shape* in the symbolic description (see figure 3.6). The meta shape describes the shape of the object in more general terms, whereby the implemented categories are: *spherical*, *cylindrical*, *cuboidal* and *other*. By applying this method, confusing two objects with the same meta shape (e.g. an apple and a tomato) is punished less than confusing two objects with different meta shapes (e.g. an apple and a bottle). In the Pick-and-Place experiments, the distance is set to 1 if the meta shapes are equal while the object shapes are not equal. This value is quite arbitrarily set, however it should be somewhere between 0 and $\sqrt{2}$ in which cases the object shapes either match or neither the object shape nor the meta shapes match between the inferred shape and the shape from the knowledge base tuple.

4. Experiments

In this chapter, the experiments will be discussed and the results will be presented. The experiments are divided into two subsections based upon the dataset that is used. That is, in section 4.1, the dataset that is used in the experiments is similar to the dataset as used by Higgins et al. [24], who originally proposed SCAN. The experiment in section 4.1 will be referred to as the *Deepmind Lab experiment*. The main purposes of the Deepmind Lab experiment are 1) to test if the neuro-symbolic model works on similar data that was used in the study that proposed SCAN and 2) to verify that the implementation of SCAN as used here is correct. This experiment is not meant to reproduce all of the results as presented by Higgins et al. [24], although the original results are used to verify the correct implementation of SCAN in this study. Next, in section 4.2, the *Pick-and-Place experiments* are presented that test the neuro-symbolic model and some iterations of the model that are trained on the pick-and-place dataset. This dataset contains manually selected images from the BigBIRD [43] and YCB Object [8] dataset and depict objects that are typically seen in pick-and-place scenarios in a warehouse setting.

4.1 Deepmind Lab experiment

4.1.1 Dataset

The dataset used in this experiment is based upon the Deepmind Lab game environment [6]. Every image in the dataset varies in terms of wall colour, floor colour, object colour and object shape (see Figure 4.1). Hereby, there exist 16 colours and 3 object shapes, resulting in a total of $16^3 \cdot 3 = 12288$ unique images. In the first experiment, about 15% of the images in dataset is selected randomly to be left out from the training data, resulting in: $n_{unseen} = 1826$ images (and thus $n_{seen} = 10462$ images). For each of the unseen images, an entry in the knowledge base is manually constructed and contains the symbolic values of wall colour, floor colour, object colour and object shape.

4.1.2 Problem definition

The problem in this experiment is defined as follows: Given dataset of 10462 training images and the knowledge base that includes a description of the unseen images, provide the image IDs of the 1826 unseen images.

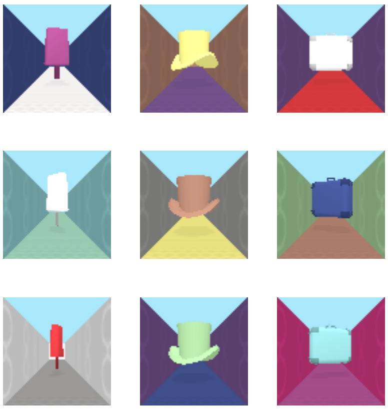


FIGURE 4.1: Some examples of the images from the Deepmind Lab dataset as used in this experiment.

4.1.3 Main Results

The neural part of the model is trained for 7 different values of the hyperparameter β. For each value of β, the percentage of correctly inferred object IDs is shown in figure 4.2.

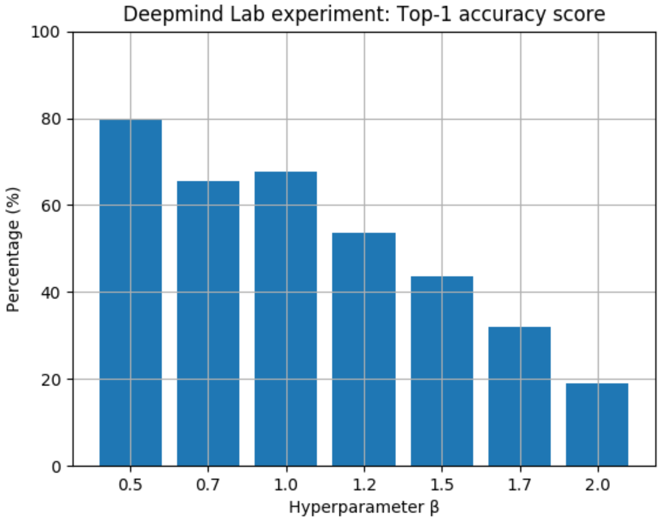


FIGURE 4.2: Top-1 accuracy score on the unseen images for different β values in the Deepmind Lab experiment.

4.1.4 Analysis

Recognizing attributes

The highest score of finding the right object ID is obtained for β value of 0.5 and between β values of 1.0 and 2.0, a constant decrease in performance can be observed. These differences in performance arise from the neural part of the model since the reasoning part is identical for each β value. Since the outcome of the neural part is a tuple of symbolic attribute values, for each of the attributes it can be analysed whether the attribute has been inferred correctly. Figure 4.3 depicts these results for all β values per attribute. As can be expected, the performance is best for a β value of 0.5 and worst for a β value of 2.0. Interestingly, the performance of the inferred object colour is relatively low compared to the other visual attributes for every value of β. A possible explanation would be that the object colour is only present in a small part of the image, thereby it contributes less to the reconstruction loss and subsequently is less prominently encoded in the latent vector. Even though, it is attempted to reduce this effect by using the reconstruction loss in feature space (see Equation 3.2), this effect might still occur. Since the goal of this experiment is to verify that the neuro-symbolic model could work, testing this hypothesis is considered beyond the scope of this experiment.

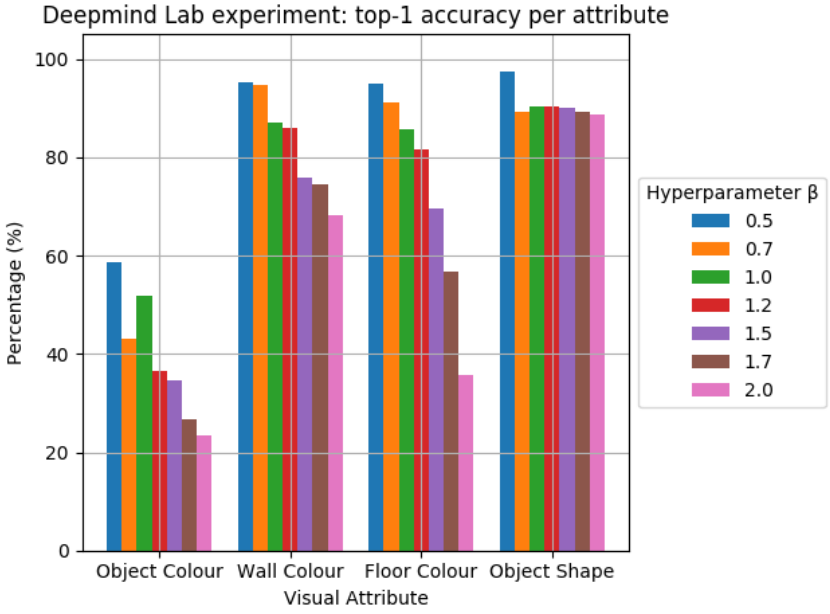


FIGURE 4.3: Top-1 accuracy score for every visual attribute of the Deepmind Lab experiment.

Disentanglement

The differences in performance can be fully ascribed to the different values of β since that's the only parameter that has been varied in this experiment. In Figure 4.4, the differences are shown between the learned latent spaces for β values of 0.5 and 2.0. Additionally, in figure 4.5, the matrices of informativeness are plotted for β values of 0.5 and 2.0.

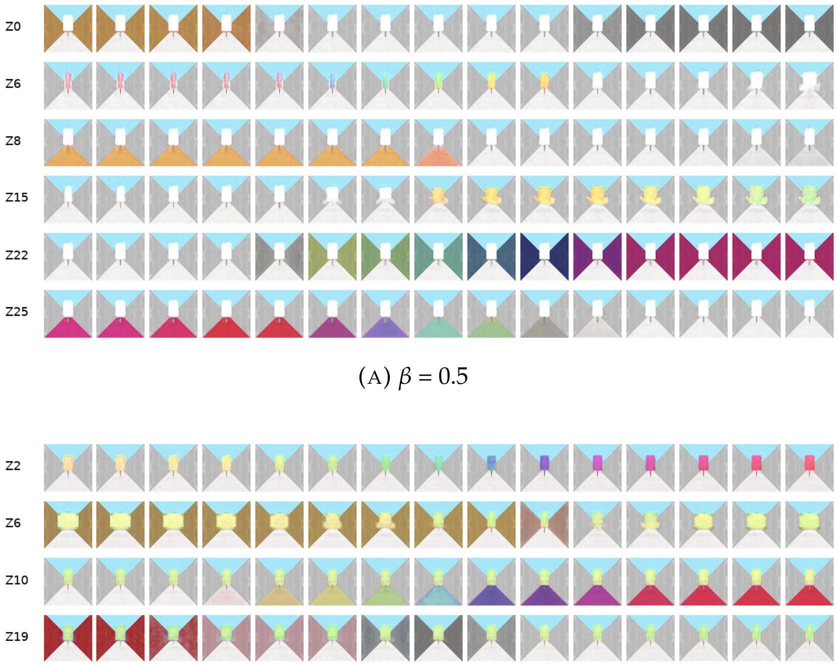
Visual analysis

As can be seen from figure 4.4, for $\beta = 0.5$, six latent factors are actively contributing to significant changes of the reference image, whereas for $\beta = 2.0$ that's the case for only four latent factors. As described in the Methods section, in this thesis, two properties are examined to assess the level of disentanglement:

1. *Independence property:* For $\beta = 0.5$, Z0 and Z22 influence wall colour. Also, Z25 and Z8 both encode floor colour and, lastly, Z15 and Z6 both cause a change of object colour. For $\beta = 2.0$, Z6 and Z19 both influence wall colour.

2. *Uniqueness property:* For both of these β values, it appears that some latent factors encode multiple visual attributes at the same time. For $\beta = 0.5$, Z6 and Z15 encode object colour and also causes a change of object shape. For $\beta = 2.0$, Z6 encodes wall colour as well as object shape.

These observations indicate that the uniqueness property is better met for $\beta = 2.0$. Next, the matrices of informativeness are analysed to provide supportive evidence for this statement.



(A) $\beta = 0.5$

(B) $\beta = 2.0$

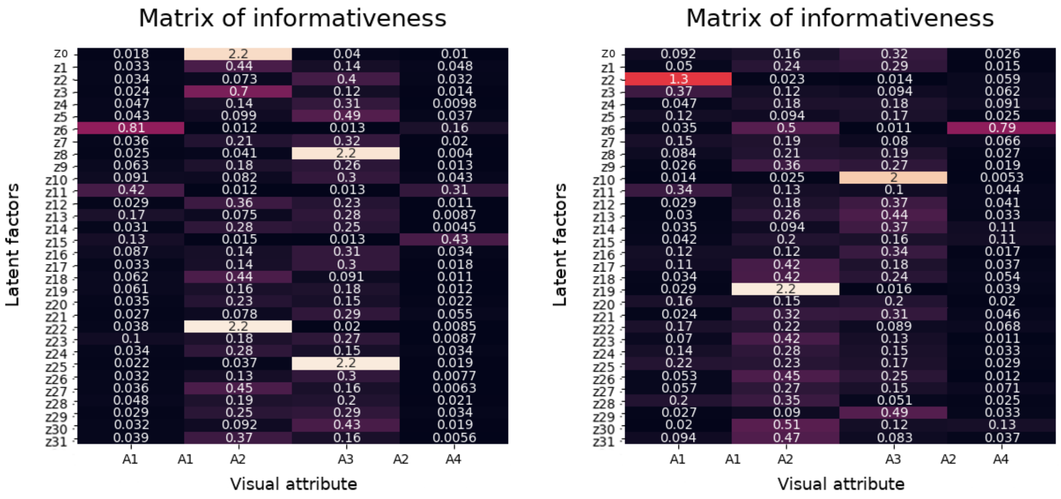
FIGURE 4.4: Visual depiction of latent spaces in experiment 1 for β values of 0.5 and 2.0. Note: Latent factors that do not encode anything visually significant are not depicted.

Matrix of informativeness
In figure 4.5, the matrices of informativeness are plotted for $\beta = 0.5$ and $\beta = 2.0$. The properties of a matrix of informativeness of a highly disentangled latent space are evaluated:

1. *Independence property:* Regarding this property, it can be seen that for $\beta = 2.0$ it is more distinctive that for every single visual attribute, a single latent factor exists with a high mutual information score compared to $\beta = 0.5$. This would suggest that for $\beta = 2.0$, a better disentanglement is achieved.

2. *Uniqueness property:* This property is not fully met for Z6 of the matrix of informativeness for $\beta = 2.0$. As can be seen in figure 4.5b, a relatively high score on mutual information is calculated for A2 and A4 (0.5 and 0.79 respectively). This would mean that latent factor Z6 encodes wall colour as well as object shape, which can be confirmed by looking at figure 4.4b. Regarding the matrix of informativeness for $\beta = 0.5$, there multiple latent factors that seem to have a relatively high mutual information score on multiple visual attributes

(e.g. Z11 and Z12). However, these latent factors are not included in figure 4.4a, which means that they do not encode something visually significant for the used reference image. Instead, given figure 4.4a one could argue that the second property for a highly disentangled latent space is not met by Z6 since this latent factor encodes object colour as well as object shape. Indeed, one could see that these visual attributes (A0 and A3) have a slightly higher score compared to the other visual attributes (A1 and A2). The same phenomena applies to Z15 for object colour and object shape (A0 and A3).



(A) $\beta = 0.5$

(B) $\beta = 2.0$

FIGURE 4.5: Matrices of informativeness for the Deepmind Lab experiment. The values on the x-axis are the visual attributes, where: A1 = object colour; A2 = wall colour; A3 = floor colour; A4 = object shape. The y-axis consists of the elements from the latent vector of the β-VAE$_d$.

Conclusion about disentanglement

Due to the imperfections in the disentanglement and lack of a decent metric to quantify disentanglement, one could argue which latent space has the best disentanglement. However, given the properties for a high disentangled latent space as defined in this study (see section 3.1.1), we conclude that the latent space is better disentangled for $\beta = 2.0$ compared to $\beta = 0.5$. Interestingly, the top-1 accuracy score is higher for $\beta = 0.5$ compared to $\beta = 2.0$.

4.2 Pick-and-Place experiments

The following set of experiments is conducted to test if the neuro-symbolic model is capable of recognizing unseen objects given images that are typically seen in pick-and-place scenarios.

4.2.1 Dataset

A training dataset and a test dataset are constructed from the images of the BigBIRD dataset [43] and YCB Object dataset [8]. In appendix A, all the objects that have been selected in this study are depicted. The training dataset consists of 44 objects and for each object 149 orientations are selected from 2 camera angles, resulting in a total of 13112 training images. The test dataset contains the images of 30 unseen objects and for each object 11 orientations have been randomly picked for 2 camera angles, resulting in a total of 660 test images. In addition to the depth images, RGB image and image mask from the BigBIRD and YCB Object dataset, the HHA encoding of the depth image [20] is constructed. This is a three layer encoding with one layer describing *horizontal disparity* (H), one layer describing *height above ground* (H) and one layer describing the *angle with gravity* (A). To analyze the performance of the model, the test dataset is split into two categories, being: *T1* that contains unicoloured object and *T2* that contains multicoloured objects (see figure 4.6).

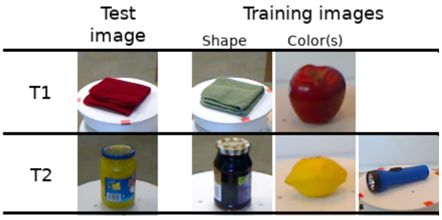


FIGURE 4.6: Examples of test images from category T1 and T2. They have an unseen combination of visual attributes, but the visual attribute values individually are present in the training dataset.

In addition to the datasets, a symbolic description is made for every object describing the object colour, object shape and the corresponding object meta shape for which has the possible values of *cuboidal*, *cylindrical*, *spherical* or *other* (see figure 3.6 for an example).

Problem definition

Given the 13112 training images with labeled attributes, provide the object IDs of the 660 unseen objects for which a symbolic description is provided. Hereby, the unseen objects have visual attributes that are present in the training data.

4.2.2 Experiment 1: RGB data

This first Pick-and-Place experiment serves as a baseline for next iterations of the neuro-symbolic model as described in the following sections. The neuro-symbolic model is trained for 9 values of hyperparameter β (i.e. $\beta \in [0.0, 0.2, 0.5, 0.7, 1.0, 1.2, 1.5, 1.7, 2.0]$). Additionally, the influence of different values of hyperparameter λ is tested for 4 values of hyperparameter β (i.e. $\lambda \in [5, 10, 15, 20]$ and $\beta \in [0.5, 1.0, 1.5, 2.0]$).

Main results

In figure 4.7, an overview of the results for every tested value of hyperparameter β is depicted. The best top-1 accuracy score of the object ID of an unseen object is 25.5% for $\beta = 0.7$. Since the test dataset on which performance analysis is done consists of 30 objects (i.e. T1 and T2 combined), a random guess would result in a percentage of correctly inferred object IDs of about 3.3%. Hence, the neuro-symbolic model is able to perform better than random guessing and is able to recognize unseen objects to some extend. Also, figure 4.7 shows that for most values of hyperparameter β, the model is able to infer the correct object ID better for unseen unicoloured objects (i.e. T1) than for unseen multicoloured objects (i.e. T2).

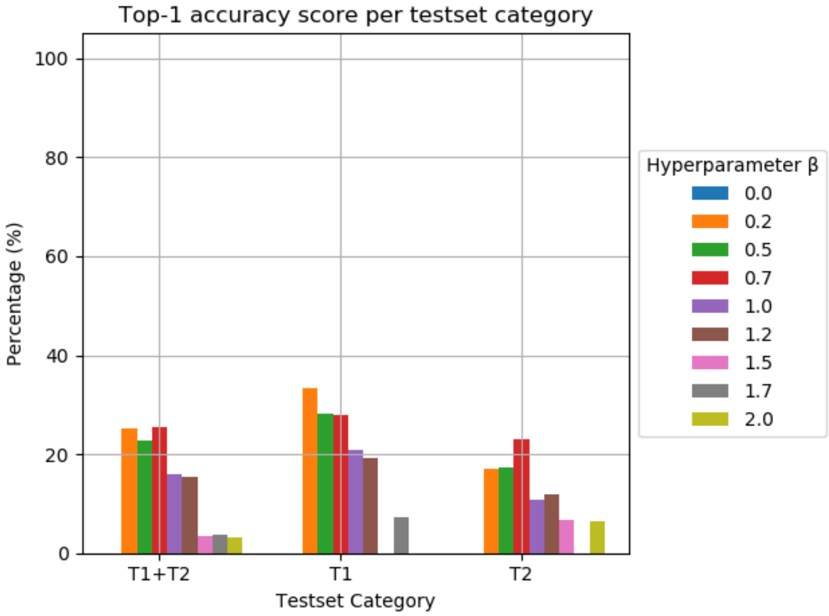


FIGURE 4.7: Overview of top-1 accuracy scores of experiment 1 of the Pick-and-Place experiments.

The influence of hyperparameter β on the performance appears to be significantly larger than the influence of hyperparameter λ on the performance in this experiment (see table 4.1). In fact, the exhibited differences in performance most likely be due to random noise rather than the different values of λ: there seems to be no consistent trend that emerges (by changing the λ value) for each of the β values.

		λ			
		5	**10**	**15**	**20**
β	**0.5**	21.3	20.1	22.2	21.1
	1.0	13.9	14.0	12.2	12.8
	1.5	0.0	0.0	2.8	2.7
	2.0	0.0	1.3	2.8	2.8

TABLE 4.1: Top-1 accuracy per value of β and λ. It can be seen that the influence of λ on the top-1 accuracy performance is limited compared to the influence of β.

Analysis

In general, a low percentage of correctly inferred object IDs could be due to one of the following reasons: 1. The neural part of the model is not able to infer a good set of attribute values or 2. the symbolic part of the model is not able to find the correct object ID or 3. a combination of these two reasons. Here, deeper analysis is conducted to gain insight on which reason(s) apply and to provide a lead for next experiments.

Recognizing attributes

In figure 4.8, the percentages of correctly inferred visual attribute values per β value are depicted. As can be seen, the percentages of correctly inferred visual attributes are about the same as the percentages of correctly inferred object IDs. This suggests that the neural part of the model is the bottleneck instead of the symbolic reasoner.

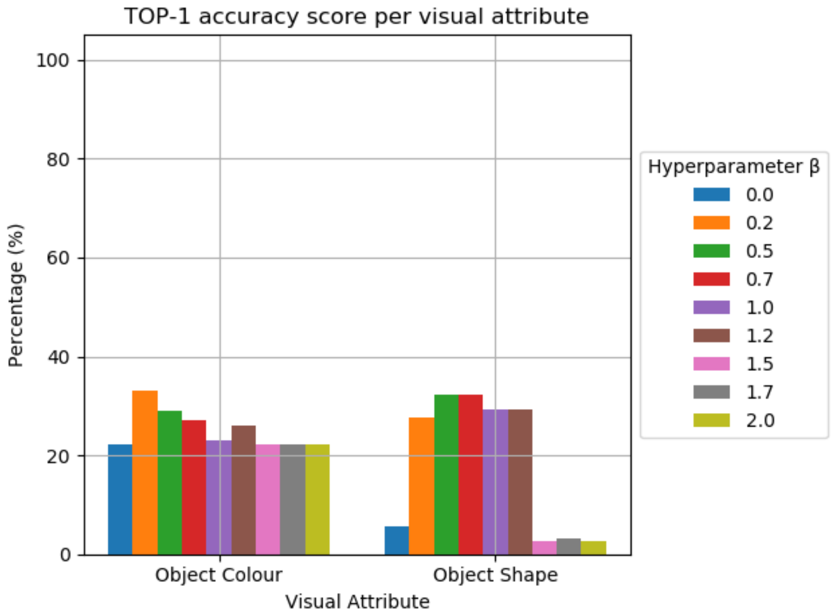


FIGURE 4.8: Top-1 accuracy scores per visual attributes for all tested values of β.

The top-1 accuracy scores per visual attributes are rather low compared to the results in the Deepmind Lab experiment (see figure 4.3). However, the neural part of the model as used in this experiment is exactly the same as for the Deepmind Lab experiment. The only factor that changed has been the training and test data.

As described in chapter 3, the neural part consists of a Denoising Autoencoder and two β Variational Autoencoders. Hereby, the DAE is being used in the loss function of the β-VAE$_d$ and, subsequently, the β-VAE$_d$ is being used in the loss function of β-VAE$_{sym}$. If the previous autoencoders is performing poorly, the autoencoders thereafter unlikely perform any better. For that reason, the performance of the DAE is evaluated qualitatively by reconstructing a certain input image and the disentanglement of the latent space of the β-VAE$_d$ is examined.

DAE performance
As can be seen from figure 4.9, the reconstructed input images from the Deepmind Lab dataset are quite crisp and only exhibit little noise. On the other hand, images from the Pick-and-Place dataset are quite noisy and, although they input images can be recognized, the reconstructed input images are not very clear.

(A) Deepmind Lab dataset

(B) Pick-and-Place dataset

FIGURE 4.9: These figures show the original image and the corresponding image as reconstructed by the DAE.

Disentanglement

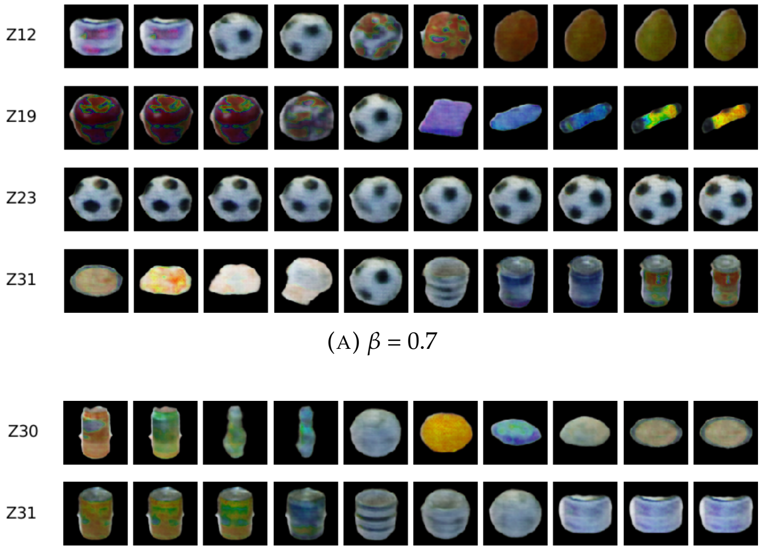
The Pick-and-Place dataset varies in different ways: object colour, object shape, object rotation and camera rotation. Ideally, if the disentanglement is high, these factors individually are encoded by a single axis in the latent space of the β-VAE$_d$. The properties of disentanglement as used in this study are examined for the β-VAE$_d$ trained for the β values 0.7 and 1.2. Figure 4.11 is used for the visual analysis and figure 4.12 depicts the corresponding matrices of informativeness.

Visual analysis
By looking at figure 4.11, it is immediately clear that the disentanglement is not as good as seen in the Deepmind Lab dataset. That is, the underlying factors of variation in the Pick-and-Place dataset (as mentioned above) are not clearly encoded by distinct latent factors. This makes the analysis quite challenging, yet an attempt is made:

1. *Independence property:* For $\beta = 0.7$, four latent factors are identified to significantly change the reference image, whereas for $\beta = 2.0$ that's the case for only two latent factors. Hereby, object shape and object colour is not clearly encoded by any of the latent factors. Although, for $\beta = 0.7$, object rotation seems to be encoded by one particular latent factor (i.e. Z23). These observations indicate that the independence property is best present for $\beta = 0.7$, although the evidence is not very strong.

2. *Uniqueness property:* For both β values, most latent factors encode object shape as well as object colour. However, for $\beta = 0.7$, Z23 seems to only encode object rotation. This would suggest that for $\beta = 0.7$ the uniqueness property is better met.

FIGURE 4.10: Reference image for analysis of latent space for the Pick-and-Place experiments.



(A) $\beta = 0.7$

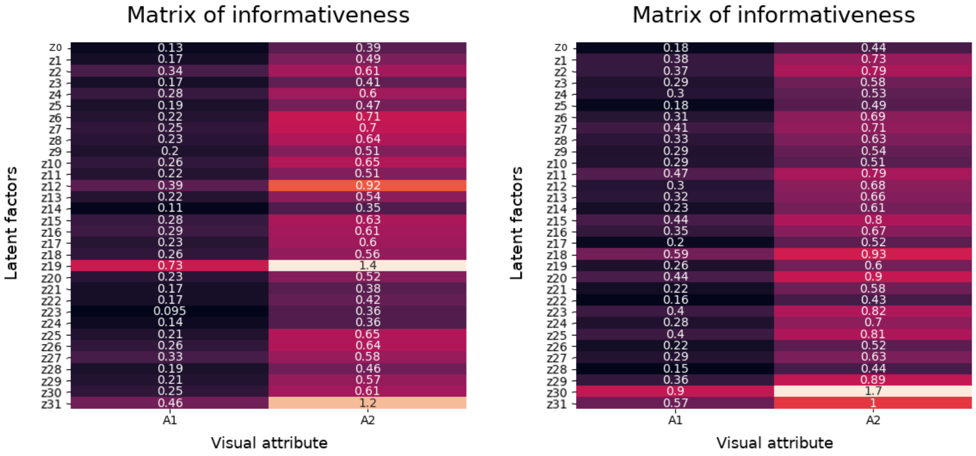
(B) $\beta = 1.2$

FIGURE 4.11: Visual depiction of the latent space for the experiment 1 for two different values of β. Note: Latent factors that do not encode anything visually significant are not depicted.

Matrix of informativeness
Along the x-axis of the matrices of informativeness, two visual attributes are represented: object colour (A1) and object shape (A2). Along the y-axis, every element from the latent vector is represented.

1. *Independence property:* For both β values, the independence property is not convincingly met for neither object colour nor object shape. That is, there is not clearly one activated latent factor within each column.

2. *Uniqueness property:* The same observation can be made for the uniqueness property: by looking at every row of the matrix of informativeness, many latent factors exist that have a high mutual information score on object shape and object colour at the same time. For example, Z19, Z27 and Z31 for $\beta = 0.7$ and Z18, Z30 and Z31 for $\beta = 1.2$.



(A) $\beta = 0.7$

(B) $\beta = 1.2$

FIGURE 4.12: Matrices of informativeness for experiment 1 of the Pick-and-Place experiments. A1 - Object colour. A2 - Object shape.

Conclusion about disentanglement

There appears to be not much of a difference in the level of disentanglement between the β-VAE$_d$ trained with β values of 0.7 or 2.0. From the visual analysis, one could argue that a slightly better disentanglement is achieved for $\beta = 0.7$ due to the existence of one latent factor that encodes object rotation only. However, object shape and object colour are quite entangled for both β values which is observed in the visual analysis as well as the matrices of informativeness. When the performance of the model is considered, the degree of disentanglement does not seem to be able to explain the differences of the top-1 accuracy between the model being trained with $\beta = 0.7$ or $\beta = 1.2$.

4.2.3 Experiment 2: RGB and depth data

Typically, a RGB-D sensor is being used to observe the scene in perception pipelines of pick-and-place systems in a warehouse setting [40]. In this experiment, the model is modified to test if the depth data enhances the performance of the model. RGB data only provides information about colour and texture. In RGB-D images, depth data is also present that provides information about geometry. It is hypothesized that the extra depth information could help to learn a mapping function that shows higher disentanglement of the latent space than if only RGB data is used. Subsequently, this could lead to an increased top-1 accuracy score of the model. In the following experiments, this hypothesis is tested. Since depth information can be handled in different ways, four modifications of the neural part of the model are tested:

Exp. 2.1 One-stream model with a RGB image with a conventional depth image concatenated.

Exp. 2.2 One-stream model with a RGB image with a HHA encoded depth image concatenated.

Exp. 2.3 Two-stream model with a RGB image and a depth image.

Exp. 2.4 Two-stream model with a RGB image and a HHA encoded depth image.

Where: the depth image is simply the preprocessed depth information as captured by the depth sensor and the HHA encoded depth image is a particular depth image encoding that has been reported to outperform conventional depth images in object recognition tasks [20]. Details about the altered neural networks can be found in the Methods section.

The neuro-symbolic model has been trained for the same set of β values as in experiment 1, being: $\beta \in [0.2, 0.5, 0.7, 1.0, 1.2, 1.5, 1.7, 2.0]$. The value of hyperparameter λ is set to $\lambda = 10$ initially since the influence of hyperparameter λ was very limited in experiment 1. In addition to the problem definition of experiment 1, in this experiment the depth or HHA image is provided.

Main results

To keep this section concise, only the results for the best values of hyperparameter β of each of the four modified models are presented. The best results are obtained for a β value of 0.2, 0.2, 0.2 and 0.5 for experiment 2.1, 2.2, 2.3 and 2.4 respectively.

As can be seen in figure 4.13, the one-stream models perform better than the baseline (i.e. the best top-1 accuracy score of experiment 1 of the Pick-and-Place experiments). The performance difference compared to the baseline is 5.3 and 5.6 percent point for the model with concatenated depth and HHA images respectively. The two-stream models only perform slightly better and show a 0.9 and 1.2 percent point increase in top-1 accuracy score compared to the baseline. Interestingly, comparing the different encodings within the one-stream and two-stream models show that for testset category T1 the regular depth encoding seems to perform better than the HHA encoding whereas for testset category T2 it's the other way around. Another result that stands out is the fact that for testset category T1 all models do perform better than the baseline model, which is not the case for testset category T2.

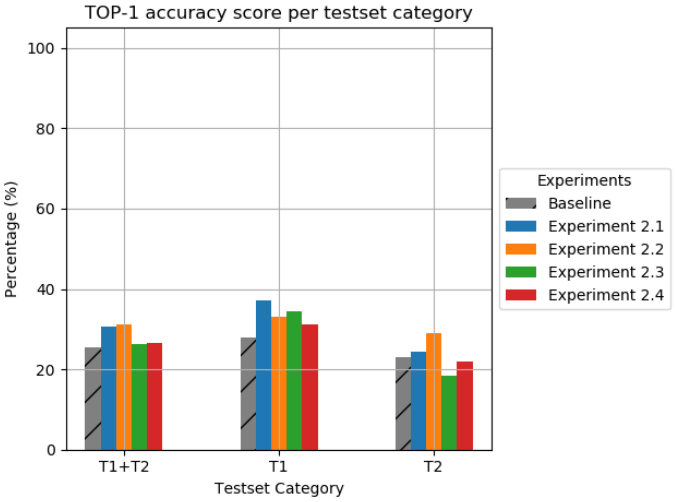


FIGURE 4.13: Top-1 accuracy results per testset category for experiments 2.1 to 2.4. The baseline score is the highest top-1 accuracy score from experiment 1 of the Pick-and-Place experiments. Only the highest scores are plotted for experiments 2.1 to 2.4.

Analysis

Compared to experiment 1 of the pick-and-place experiments, the overall performance is improved for all the tested models. Hence, based upon these results, adding depth information to the model is considered favorable. However, it must be stated that the improvements in top-1 accuracy score are rather small for experiments 3.3 and 3.4. It is stated above that providing additional depth information to the model could lead to a better disentangled latent space. In this part, it is analyzed whether this hypothesis is true or not.

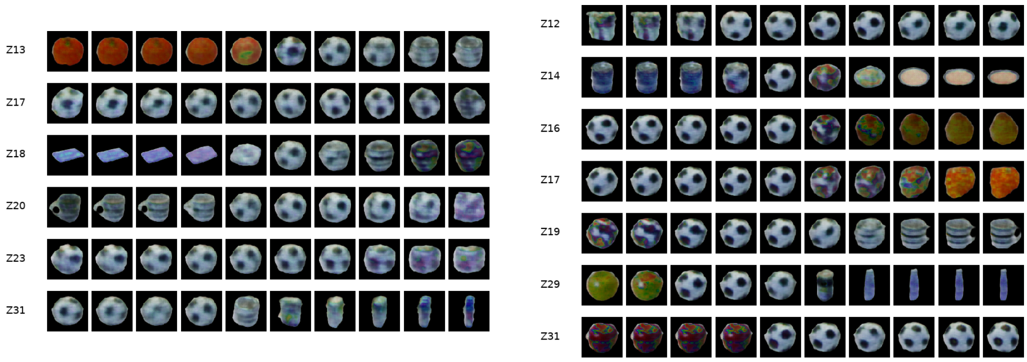
Disentanglement

By looking at the disentanglement data, the hypothesis that depth information increases the disentanglement is checked. Again, the properties of independence and uniqueness are evaluated.

Visual analysis

1. *Independence property:* From figure 4.14, it becomes immediately clear that this property does not apply very well to the latent spaces of both experiment 2.2 and 2.3. For example, the visual attribute object shape is encoded by multiple latent factors as well as object rotation for both experiments.

2. *Uniqueness property:* Also, the uniqueness property is not met in the shown visual representation of the latent spaces of experiment 2.2 nor 2.3. For example, many latent factors in both experiments cause a change in object colour and object shape at the same time.



(A) Experiment 2.2; $\beta = 0.2$

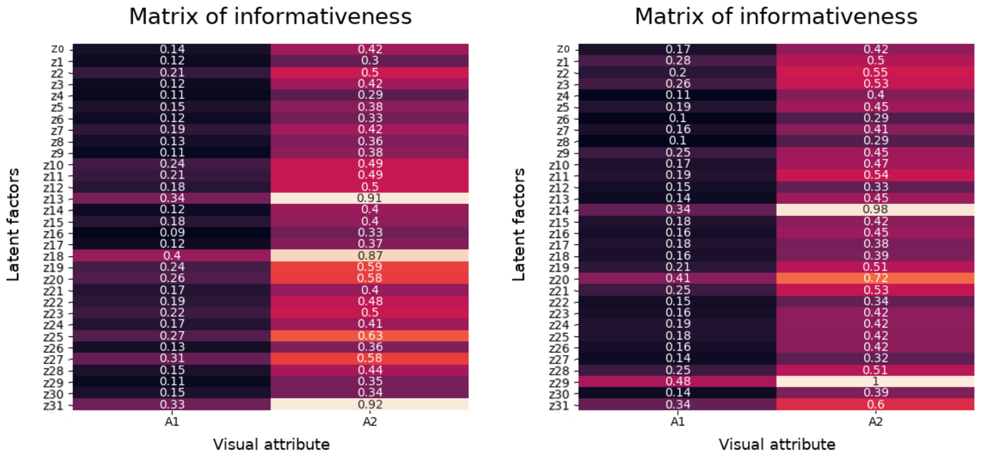
(B) Experiment 2.3; $\beta = 0.2$

FIGURE 4.14: Visual depiction of latent spaces in experiment 2.2 and 2.3 for β values of 0.2.

Matrix of informativeness
The matrices of informativeness are depicted in figure 4.15.

1. *Independence property:* The matrices of informativeness confirm the observations from the visual analysis: the independence property is not well met for neither the matrix of informativeness regarding experiment 2.2 nor experiment 2.3. In fact, the matrices of informativeness of experiment 2.2 and 2.3 look quite similar to the ones from experiment 1 (see figure 4.12).

2. *Uniqueness property:* Also the uniqueness property is not well matched in neither experiment 2.2, which can be seen by looking at Z13, Z18, Z27, nor experiment 2.3, which can be seen by looking at Z14, Z20, Z29.



(A) Experiment 2.2; $\beta = 0.2$

(B) Experiment 2.3; $\beta = 0.2$

FIGURE 4.15: Matrices of informativeness.

Conclusion about disentanglement
The level of disentanglement of the latent spaces as examined here is not very high. Neither the independence property nor the uniqueness property is well observable. In the beginning of this section, a hypothesis was posed that the extra depth information potentially leads to a higher disentangled latent space. The obtained results provide no support for this hypothesis. Therefor, the hypothesis is rejected.

4.2.4 Experiment 3: Next iterations

In this experiment, iterations of experiment 1 will be conducted to test if the model could reach or approach a performance level as seen in the Deepmind Lab experiment. The following iterations are tested:

Exp. 3.1 the same model as presented in experiment 1 with unmasked images.

Exp. 3.2 the same images as in experiment 1 with a low capacity network.

Exp. 3.3 the same images as in experiment 1 with a high capacity network.

Exp. 3.4 the same images as in experiment 1 without using the DAE.

The details of the iterations are given in the Methods section. The first iteration is motivated by the hypothesis that imperfect masks might cause a significant drop in performance by including pixels that don't belong to the object or masking out pixels that do belong to the object (see figure 4.16). The second and third iteration follow from the conclusions of previous studies that describe the importance of capacity of a neural network [5, 46]. Hereby, a straightforward explanation of the capacity of a neural network is how much information a neural network is able to encode. In experiment 1, the same model is used as in the Deepmind Lab experiment, thus the model had the same capacity. However, the dataset has changed and thereby the information present in the dataset. This might lead to the capacity of the model not being sufficient, which subsequently could contribute to the a drop in the performance of the model. Therefore, a low capacity and a high capacity neural network has been created and tested (see section 3). Lastly, the fourth iteration is made based upon the observation that the Denoising Autoencoder wasn't capable of reconstructing high quality images in experiment 1 compared to the Deepmind Lab experiment. Instead of implementing the reconstruction error in the latent space, the performance of the model is tested by taking the originally proposed reconstruction error in pixel space (see equation 3.5).

(A) (B)

FIGURE 4.16: The masks as used in the Pick-and-Place dataset are noisy to some extend. Here, an extreme example is depicted. In experiment 3.1, an effort is made to investigate the effects of this noise on the performance. In all other experiments, the masked images are being used.

Main results

To keep this section concise, only the results for the best values of hyperparameter β of each of the four modified models will be presented. The best results are obtained for a β value of 0.7, 0.7, 0.7 and 0.5 for experiment 3.1, 3.2, 3.3 and 3.4 respectively.

From figure 4.17 it can be seen that again all iterations have a better overall performance. The differences compared to the baseline experiment in top-1 accuracy score range from 1.1 percent point for experiment 3.1 to 7.0 percent point for experiment 3.4. Using the unmasked training dataset (i.e. experiment 3.1), the increase in

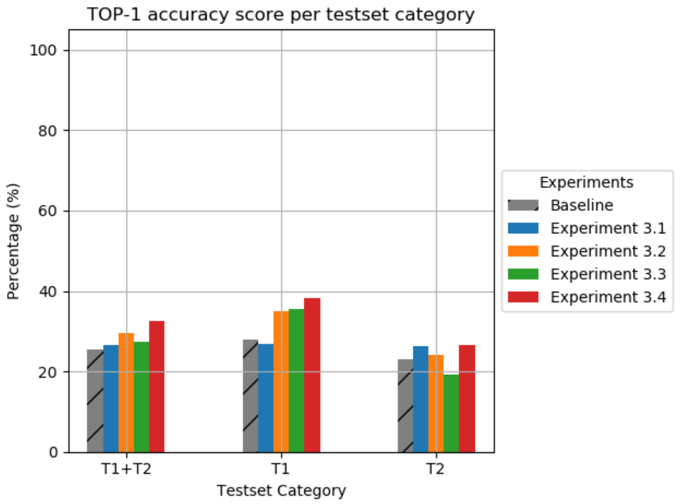


FIGURE 4.17: Top-1 accuracy results per testset category for experiments 3.1 to 3.4. The baseline score is the highest top-1 accuracy score from experiment 1 of the Pick-and-Place experiments. Only the highest scores are plotted for experiments 3.1 to 3.4.

performance is only 1.1 percent point compared to the baseline experiment. Next, for experiments 3.2 and 3.3, the top-1 accuracy scores are 4.1 and 2.0 percent point higher than the baseline experiment. Hereby, the differences of the overall score is mainly caused by better performance on the images from testset category T1. For experiment 3.3, the performance on testset category T2 is 3.6 percent point below the baseline score, which caused the overall score to drop by about 1.8 percent point. Lastly, the highest performance is achieved by replacing the reconstruction error in DAE latent space by the reconstruction error in pixel space, which is tested in experiment 3.4. The overall top-1 accuracy score of experiment 3.4 is 6.9 percent point higher than in the baseline experiment. More specifically, the differences with respect to the baseline experiment are 10.3 and 3.6 percent point for the testing categories T1 and T2 respectively.

Analysis

In experiment 3.1, the hypothesis is tested whether the inaccuracies in the masks as being used in the baseline experiment could explain (a part of) the performance differences between the Deepmind Lab experiment and the Pick-and-Place baseline experiment. The overall result show only a small difference of 1.1 percent point. However, by looking at the testset categories individually, a small drop in performance occurs for T1 (-1.2 percent point) and an improvement of performance is shown for T2 (3.3 percent point). The inaccuracies of the masks particularly occurred in the images that belong to the T2 category. This might explain why the performance rises in the T2 category. However, the quality of the masks is not measured objectively in this study since it is considered beyond the scope of this study. It is concluded that the effects of the inaccuracies of the masks on the top-1 accuracy score are relatively small.

To test the influence of the capacity of the neural network on the performance of the model, a high and low capacity model are trained and tested in experiment 3.2 and 3.3 respectively. Interestingly, the overall performance improved for both iterations compared to the baseline experiment. For category T1 especially, the low and high capacity models clearly show a better performance than in the baseline experiment. If the capacity of the neural network is due to the capacity being too low, one would expect an increase in performance from the low to the high capacity model. Additionally, for category T2, there is no clear improvement in performance for the low and high capacity model. In fact, the high capacity model even performs worse than the baseline experiment. These results suggest that the capacity of the β-VAE$_d$ does influence the performance, but it is unlikely that the capacity of the baseline model is insufficient.

Lastly, from the results of experiment 3.4, it can be stated that taking the reconstruction loss in the latent space of the Denoising Autoencoder results in a drop of performance given the Pick-and-Place problem definition. For the categories T1 and T2, the model used in experiment 3.4 scored better than the baseline experiment. As was shown in experiment 1, the DAE is not able to construct very high quality reconstructions of the input image. This indicates that substantial information is lost in the encoding process. It is presumed that by directly taking the reconstruction loss in pixel space instead (of in the latent space of the DAE), the β-VAE$_d$ is better able to learn a higher quality encoding describing the input image. In essence, as applied here, taking the reconstruction loss in the DAE latent space adds extra noise to the input image which is unfavourable.

5. Conclusions

5.1 Summary of results

In this study, a neuro-symbolic model is applied to solve a zero-shot learning problem in the domain of object recognition in a Pick-and-Place scenario. In table 5.1, an overview is provided of the best scores for each of the experiments. The results show that the proposed model could achieve a top-1 accuracy score of 79.5% on the unseen images of the Deepmind Lab dataset. In the baseline experiment of the Pick-and-Place scenario (experiment 1), the top-1 accuracy score was 25.5%. In the next experiments, some iterations of the model were proposed, whereby the best achieved top-1 accuracy score on the unseen images was 32.4%.

Dataset	Experiment	Best top-1 accuracy score	β value	Properties of experiment
Deepmind Lab	-	79.5%	0.5	Baseline model + Deepmind Lab dataset
Pick-and-Place	Experiment 1	25.5%	0.7	Baseline model + Pick-and-Place datset
Pick-and-Place	Experiment 2.1	30.8%	0.2	One-stream + Depth
Pick-and-Place	Experiment 2.2	31.1%	0.2	One-stream + HHA
Pick-and-Place	Experiment 2.3	26.4%	0.2	Two-stream + Depth
Pick-and-Place	Experiment 2.4	26.7%	0.5	Two-stream + HHA
Pick-and-Place	Experiment 3.1	26.5%	0.7	Baseline model + No masks
Pick-and-Place	Experiment 3.2	29.5%	0.7	Low capacity neural network
Pick-and-Place	Experiment 3.3	27.4%	0.7	High capacity neural network
Pick-and-Place	Experiment 3.4	32.4%	0.5	Loss function in pixel space

TABLE 5.1: Overview of the best top-1 accuracy scores on unseen images for all the experiments in this thesis. The β value corresponds to the reported top-1 accuracy score.

5.2 Answers to the research questions

The main research question in this thesis is:

Can a neuro-symbolic model be used to recognize unseen objects given RGB-D data as typically seen in pick-and-place scenarios?

The highest top-1 accuracy score of recognizing unseen images of all the Pick-and-Place experiments was 32.4%. Compared to other studies focusing on the same problem, this score is rather low. That is, Zeng et al. [52] reported a top-1 accuracy score on unseen images of 82.1%. However, their score is based upon their experiment, which is different from the experiments as conducted in this study: for each of the unseen test images there were 20 possible object IDs, whereas in this study there where 36 possible object IDs. Hence, comparing the results from the study of Zeng et al. [52] with the results presented here likely results in a bias in favour of Zeng et al. [52]. Nonetheless, the performance difference is substantial and it is likely that the framework as proposed by Zeng et al. [52] would outperform the neuro-symbolic model as proposed here in a fair comparison. Therefore, we conclude that the neuro-symbolic model as proposed here is not suitable in its current form to recognize unseen objects in a pick-and-place scenario.

5.2.1 Subquestions

1. *Is the neuro-symbolic model capable of recognizing unseen images of the Deepmind Lab dataset?*

   The short answer to the subquestion would be that the neuro-symbolic model is capable of recognizing unseen images of the Deepmind Lab dataset up to 79.5%, which is considered successful. However, as described in the introduction, the motivation behind this question was twofold: 1) testing the neuro-symbolic model using the Deepmind Lab dataset eliminates a potential performance drop due to the differences between the Deepmind Lab dataset and the Pick-and-Place dataset. And 2) to test the implementation of SCAN since the original code base is not publicly available. These two points will shortly be discussed next.

   Regarding the first point, in the Deepmind Lab experiment, the best top-1 accuracy score was 79.5% on the unseen images. Unfortunately, a large performance drop occurs between the Deepmind Lab experiment and the Pick-and-Place experiments. That is, the difference between the best top-1 accuracy score of the Deepmind Lab experiment (i.e. 79.5%) and the Pick-and-Place experiments (i.e. 32.4%) is 47.1 percent point. Many reasons may cause this difference in performance. An obvious reason would be that the neuro-symbolic model is not capable of bridging the reality gap. This is a common problem in computer vision and means that the performance of a certain neural network drops significantly when real data is used instead of synthetic data. Although it is very likely that this effect occurred in this study, there might be other reasons as well. In experiment 3, it was hypothesized that noisy object masks might drop the performance of the network. However, this hypothesis was rejected after measuring about the same top-1 accuracy score on unseen images without an object mask in experiment 3.1. Suggestions for further research to pinpoint the reasons behind this large difference are provided in the future work section below.

   Regarding the second point, the learned latent spaces were quite similar to the results as reported by Higgins et al. [24]. That is, for particular values of β, a high disentanglement was achieved whereby floor colour, wall colour, object colour and object shape were largely independently encoded in the latent space (as can be seen in figure 4.4). The results of neural part achieved in the Deepmind Lab experiment were considered good enough to continue developing the neuro-symbolic model and test it on unseen images.

2. *What is the influence of disentanglement on the top-1 accuracy of recognizing unseen images?*
   In the literature, it has been reported that a high disentangled latent space is desirable since it increases the explainability of the model [11, 18] and has been shown to lead to better performances [24]. However, there is no clear consensus about the definition of disentanglement, making it hard to tell whether or not a particular latent space has high disentanglement or not [36, 44]. In this thesis, two properties have been defined that a highly disentangled latent space should have: the *independence* property (i.e. every visual attribute is maximally encoded by one latent vector) and the *uniqueness* property (i.e. a particular latent vector maximally encodes one visual attribute). To assess the level of disentanglement, these properties were checked by visual analysis and by examining the matrix of informativeness. The latent spaces for the

Pick-and-Place experiments were rather entangled. On the other hand, the latent spaces as seen in the Deepmind Lab experiment exhibited higher levels of disentanglement. Given the fact that significantly higher performance is achieved in the Deepmind Lab experiment, it is argued that disentanglement indeed has a significant effect on the top-1 accuracy. At the same time, the largely entangled latent spaces as seen in the Pick-and-Place experiments likely caused the low top-1 accuracy scores. However, in the analysis part of the Deepmind Lab experiment, it was concluded that the latent space for $\beta = 2.0$ was better disentangled than for $\beta = 0.5$, while for $\beta = 0.5$ the highest top-1 accuracy score was achieved. A possible explanation could be that the property of uniqueness actually lowers the top-1 accuracy score and that multiple latent factors encoding one particular attribute is actually favorable due to the increased possibilities for (fine)tuning an object colour or object shape. Concluding, the results provide evidence that the level of disentanglement is a key factor in the performance of the neuro-symbolic network. Yet, higher levels of disentanglement (as measured in this study) do not necessarily lead to higher top-1 accuracy scores.

3. *Does depth information increase the top-1 accuracy of recognizing unseen objects?*
In experiment 2 of the Pick-and-Place experiments, it was shown that the neuro-symbolic model with an one-stream neural network performed better than the baseline model and the model with the two-stream neural network. Two depth encodings were tested: plain depth data as captured from the RGB-D camera and the HHA encoded depth data. For both of these encodings, the top-1 accuracy scores of the one-stream neural networks were about 5.3 and 5.6 percent point better than the baseline model and 4.4 and 4.4 percent point better compared to the model with the two-stream neural networks. It was hypothesized that the extra depth data might increase the level of disentanglement of the latent space. However, the results do not support this hypothesis. Thus, it is yet unclear what drives the increase of the top-1 accuracy score.

4. *How does the capacity of the neural network(s) influence the performance of the neuro-symbolic model?*
In experiment 3.2 and 3.3, the performance of respectively a low and a high capacity version of the neural part of the model was assessed. Remarkably, both the low and high capacity model scored a higher top-1 accuracy score than in the baseline experiment. These results suggest that the the capacity of the neural networks does play a role in the performance, but that the performance of the neuro-symbolic model in the baseline experiment was not limited by the capacity being insufficient. After all, in that case, one would expect the top-1 accuracy to increase from the low capacity experiment to the baseline experiment and to the high capacity experiment.

5. *Can the loss function as used in SCAN be modified to achieve a better performance of the neuro-symbolic model?*
Higgins et al. [23, 24] have proposed to define the reconstruction error of the β-VAE$_d$ in the latent space of a DAE rather than in pixel space. This was done so that small, but important features in the image would contribute more to the reconstruction error. Ultimately, it has been reported to result in a better representation of small features in the latent space of the β-VAE$_d$ [23]. However, in experiment 1 of the Pick-and-Place experiments, it is seen that the reconstructed images from the DAE appear to suffer from significant quality loss. It

is hypothesized that the performance of the model would actually be diminished by defining the reconstruction error in the latent space of the DAE rather than be improved. Instead, in experiment 3.4 of the Pick-and-Place experiments, the reconstruction loss was defined in pixel space. The best result of experiment 3.4 was a top-1 accuracy score of 32.4%, which is 6.9 percent point higher than the baseline experiment. Therefor, the aforementioned hypothesis is accepted. Concluding, changing the loss function as used in SCAN leads to a better performance of the neuro-symbolic model by defining the reconstruction error in pixel space instead of the latent space of a DAE.

5.3 Limitations

Several limitations are imposed by the proposed model and the experimental design.

First, the assumption is made that object colour and object shape are sufficient to identify the unseen object. This might not be the case. For example, there might be one light blue cup and one dark blue cup among the unseen objects. This would both lead to the same entry in the knowledge base as used in this study. Only if the visual attributes can be recognized more specifically by the neural part of the model and are more specifically described in the knowledge base, this problem could potentially be solved. Additionally, one might want to include more information about the object. Using the cup example, extra information could be a binary value that encodes if the cup has a handle or not.

Second, the images as used in the Pick-and-Place experiment depict non occluded, single objects. Most often, however, this is not the case for real-world Pick-and-Place scenarios. To make the neuro-symbolic model suitable for cluttered scenes with occluded objects, the framework has to be extended. For example with an object agnostic segmentation method [16] or by using an object agnostic grasping mechanism to apply the *grasp-first-then-recognize* paradigm as proposed by Zeng et al. [52].

Lastly, the object shape and object colour had to be present in the training data. For example, if the neural network never had been trained on yellow objects, it won't be able to recognize yellow as an object colour. This assumption also applies to shapes, which could be problematic if different shapes are too dissimilar. For example, there are many different shapes of cups, but the model will only be able to recognize an unseen cup if the shape is quite similar to the shapes within the training dataset.

5.4 Future work

Although the performance of the neuro-symbolic model was not very high for the Pick-and-Place experiments, the Deepmind Lab experiment shows it is possible to achieve a high performance. Further research is needed to investigate how the neuro-symbolic model could be improved and what its limitations are. The following suggestions are made for future studies:

- *Test the neuro-symbolic model on a synthetic dataset similar to the Pick-and-Place dataset* - The differences in performance between the Deepmind Lab experiment and the Pick-and-Place experiments are likely (to a certain extend) explained by the fact that the Deepmind Lab dataset is synthetic dataset whereas the Pick-and-Place dataset consists of real images. However, that's not the only difference: The Deepmind Lab dataset consists of 12288 images whereby each image has a unique combination of wall colour, object colour, floor colour and object shape. On the other hand, the Pick-and-Place dataset consists 13112 images and includes 149 different object orientations from 2 camera angles for 44 objects. These different properties of the datasets might also contribute to the differences in performance. By creating a synthetic dataset that has the same properties as the Pick-and-Place dataset, it can be assessed what are the most significant factors underlying the performance.

- *Compare the neuro-symbolic model with other zero-shot learning frameworks in a fair comparison study -* In this thesis, the main goals was to find out if a neuro-symbolic model would be able to recognize unseen images in a pick-and-place scenario. Therefor, a dataset was constructed that consists of images that are normally seen in the considered application. However, by using this custom dataset, the results from this study can not fairly be compared with other zero-shot learning frameworks. Xian et al. [50] have proposed a benchmark to compare different zero-shot learning frameworks and thus it would be valuable to run the neuro-symbolic model on this benchmark.

- *Implement different fusion methods for the two-stream architecture -* In experiment 2 of the Pick-and-Place experiments, depth data was additionally provided to the neural part of the model. Whereas for the one-stream architecture better top-1 accuracy scores were achieved compared to baseline, for the two-stream architecture the top-1 score was comparable to the top-1 accuracy score of the baseline. However, the used method to fuse the depth encoding and the RGB encoding was very simple. In the literature, more sophisticated fusion methods have been proposed [17, 34]. Exploring different fusion methods might cause a higher top-1 accuracy performance for the two-stream architecture and is suggested to explore in future work.

A. Pick-and-Place Experiments

A.1 Dataset

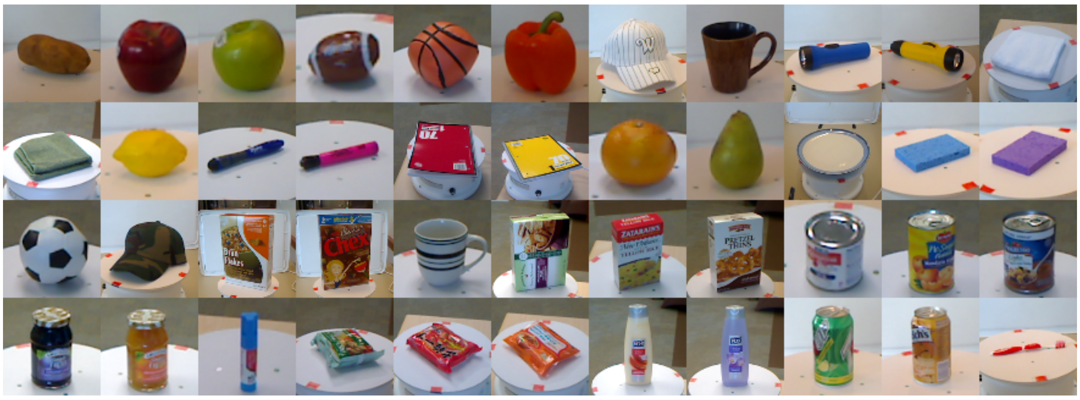


FIGURE A.1: Sample of each object from training dataset of pick-and-place scenario.

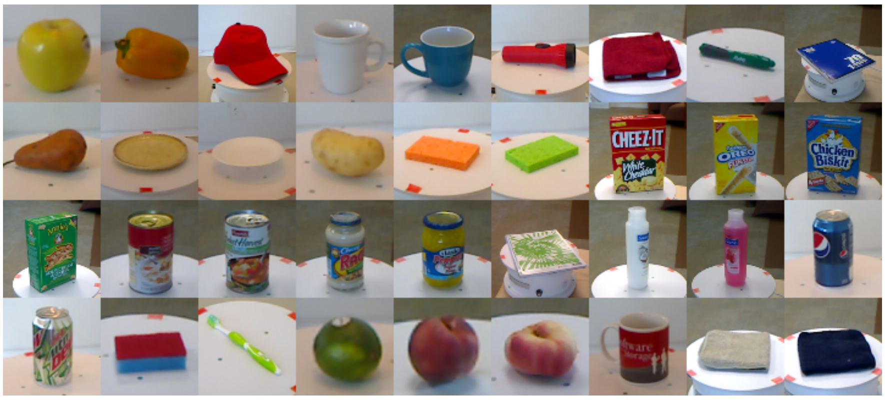


FIGURE A.2: Sample of each object from test dataset of pick-and-place scenario.

A. Bibliography

[1] *2015 Amazon Picking Challenge - Overview*. URL: https://www.amazonrobotics.com/site/binaries/content/assets/amazonrobotics/pdfs/2015-apc-summary.pdf.

[2] *2016 - APC Rules*. URL: https://www.amazonrobotics.com/site/binaries/content/assets/amazonrobotics/pdfs/2016-amazon-picking-challenge---official-rules.pdf.

[3] *2017 Amazon Robotics Challenge - Official Rules*. URL: https://www.amazonrobotics.com/site/binaries/content/assets/amazonrobotics/arc/2017-amazon-robotics-challenge-rules-v3.pdf.

[4] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation". In: *arXiv:1511.00561 [cs]* (Oct. 10, 2016). arXiv: 1511.00561. URL: http://arxiv.org/abs/1511.00561 (visited on 07/01/2020).

[5] Eric B Baum. "On the capabilities of multilayer perceptrons". In: *Journal of Complexity* 4.3 (1988), pp. 193–215. ISSN: 0885-064X. DOI: https://doi.org/10.1016/0885-064X(88)90020-9. URL: https://www.sciencedirect.com/science/article/pii/0885064X88900209.

[6] Charles Beattie et al. "DeepMind Lab". In: *CoRR* abs/1612.03801 (2016). arXiv: 1612.03801. URL: http://arxiv.org/abs/1612.03801.

[7] Yoshua Bengio. "Deep Learning of Representations: Looking Forward". In: *CoRR* abs/1305.0445 (2013). arXiv: 1305.0445. URL: http://arxiv.org/abs/1305.0445.

[8] B. Calli et al. "The YCB object and Model set: Towards common benchmarks for manipulation research". In: *2015 International Conference on Advanced Robotics (ICAR)*. 2015, pp. 510–517.

[9] A. Causo et al. "A Robust Robot Design for Item Picking". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. 2018 IEEE International Conference on Robotics and Automation (ICRA). May 2018, pp. 7421–7426. DOI: 10.1109/ICRA.2018.8461057.

[10] Tian Qi Chen et al. "Isolating Sources of Disentanglement in Variational Autoencoders". In: *CoRR* abs/1802.04942 (2018). arXiv: 1802.04942. URL: http://arxiv.org/abs/1802.04942.

[11] Xi Chen et al. "InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets". In: *CoRR* abs/1606.03657 (2016). arXiv: 1606.03657. URL: http://arxiv.org/abs/1606.03657.

[12] Xinlei Chen, Abhinav Shrivastava, and A. Gupta. "NEIL: Extracting Visual Knowledge from Web Data". In: *2013 IEEE International Conference on Computer Vision* (2013), pp. 1409–1416.

[13] Carlos Hernández Corbato et al. "Integrating different levels of automation: Lessons from winning the amazon robotics challenge 2016". In: *IEEE Transactions on Industrial Informatics* 14.11 (2018), pp. 4916–4926.

[14] Cian Eastwood and Christopher KI Williams. "A framework for the quantitative evaluation of disentangled representations". In: *International Conference on Learning Representations*. 2018.

[15] Philipp Fischer. "Descriptor Matching with Convolutional Neural Networks: a Comparison to SIFT". In: (), p. 11.

[16] Wouter Van Gansbeke et al. *Unsupervised Semantic Segmentation by Contrasting Object Mask Proposals*. 2021. arXiv: 2102.06191 [cs.CV].

[17] M. Gao et al. "RGB-D-Based Object Recognition Using Multimodal Convolutional Neural Networks: A Survey". In: *IEEE Access* 7 (2019), pp. 43110–43136. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2907071.

[18] Marta Garnelo and Murray Shanahan. "Reconciling deep learning with symbolic artificial intelligence: representing objects and relations". In: *Current Opinion in Behavioral Sciences*. SI: 29: Artificial Intelligence (2019) 29 (Oct. 1, 2019), pp. 17–23. ISSN: 2352-1546. DOI: 10.1016/j.cobeha.2018.12.010. URL: http://www.sciencedirect.com/science/article/pii/S2352154618301943 (visited on 07/02/2020).

[19] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *Entropy and Mutual Information*. Amherst, Massachusetts: University of Massachusetts, 2013.

[20] Saurabh Gupta et al. "Learning Rich Features from RGB-D Images for Object Detection and Segmentation". In: *CoRR* abs/1407.5736 (2014). arXiv: 1407.5736. URL: http://arxiv.org/abs/1407.5736.

[21] Stevan Harnad. "The Symbol Grounding Problem". In: *CoRR* cs.AI/9906002 (1999). URL: http://arxiv.org/abs/cs.AI/9906002.

[22] Irina Higgins et al. "beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework". In: *ICLR*. 2017.

[23] Irina Higgins et al. "DARLA: Improving Zero-Shot Transfer in Reinforcement Learning". In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, June 2017, pp. 1480–1490. URL: http://proceedings.mlr.press/v70/higgins17a.html.

[24] Irina Higgins et al. "Scan: Learning hierarchical compositional visual concepts". In: *arXiv preprint arXiv:1707.03389* (2017).

[25] Elad Hoffer, Itay Hubara, and Nir Ailon. "Deep unsupervised learning through spatial contrasting". In: *arXiv:1610.00243 [cs, stat]* (Oct. 2, 2016). arXiv: 1610.00243. URL: http://arxiv.org/abs/1610.00243 (visited on 08/07/2019).

[26] Phillip Isola, Joseph J. Lim, and Edward H. Adelson. "Discovering States and Transformations in Image Collections". In: *CVPR*. 2015.

[27] Dinesh Jayaraman and Kristen Grauman. "Zero Shot Recognition with Unreliable Attributes". In: *CoRR* abs/1409.4327 (2014). arXiv: 1409.4327. URL: http://arxiv.org/abs/1409.4327.

[28] Rico Jonschkowski et al. "Probabilistic Multi-Class Segmentation for the Amazon Picking Challenge". In: (2016), p. 8.

[29] Diederik P Kingma and Max Welling. *Auto-Encoding Variational Bayes*. 2013. arXiv: 1312.6114 [stat.ML].

[30] Thomas N Kipf and Max Welling. "Semi-supervised classification with graph convolutional networks". In: *arXiv preprint arXiv:1609.02907* (2016).

[31] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[32] R. Kumar et al. "Object detection and recognition for a pick and place Robot". In: *Asia-Pacific World Congress on Computer Science and Engineering*. 2014, pp. 1–7.

[33] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. "Learning to detect unseen object classes by between-class attribute transfer". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 951–958. DOI: 10.1109/CVPR.2009.5206594.

[34] H. Liu et al. "Weakly Paired Multimodal Fusion for Object Recognition". In: *IEEE Transactions on Automation Science and Engineering* 15.2 (Apr. 2018), pp. 784–795. ISSN: 1558-3783. DOI: 10.1109/TASE.2017.2692271.

[35] Francesco Locatello et al. "On the Fairness of Disentangled Representations". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 14611–14624. URL: http://papers.nips.cc/paper/9603-on-the-fairness-of-disentangled-representations.pdf.

[36] Francesco Locatello et al. "On the Fairness of Disentangled Representations". In: *CoRR* abs/1905.13662 (2019). arXiv: 1905.13662. URL: http://arxiv.org/abs/1905.13662.

[37] Eiichi Matsumoto et al. "End-to-End Learning of Object Grasp Poses in the Amazon Robotics Challenge". In: (2017), p. 4.

[38] A. Milan et al. "Semantic Segmentation from Limited Training Data". In: *arXiv:1709.07665 [cs]* (Sept. 22, 2017). arXiv: 1709.07665. URL: http://arxiv.org/abs/1709.07665 (visited on 07/01/2019).

[39] Muhammad Ferjad Naeem et al. "Learning Graph Embeddings for Compositional Zero-shot Learning". In: *CoRR* abs/2102.01987 (2021). arXiv: 2102.01987. URL: https://arxiv.org/abs/2102.01987.

[40] N.M. van der Sar. "A review on visual perception in pick-and-place robots". In: (2020).

[41] Max Schwarz et al. "Fast Object Learning and Dual-arm Coordination for Cluttered Stowing, Picking, and Packing". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. 2018 IEEE International Conference on Robotics and Automation (ICRA). Brisbane, QLD: IEEE, May 2018, pp. 3347–3354. ISBN: 978-1-5386-3081-5. DOI: 10.1109/ICRA.2018.8461195. URL: https://ieeexplore.ieee.org/document/8461195/ (visited on 07/01/2019).

[42] Anna Sepliarskaia, Julia Kiseleva, and Maarten de Rijke. *Evaluating Disentangled Representations*. 2019. arXiv: 1910.05587 [cs.LG].

[43] Arjun Singh et al. "Bigbird: A large-scale 3d database of object instances". In: *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2014, pp. 509–516.

[44] Sjoerd van Steenkiste et al. "Are Disentangled Representations Helpful for Abstract Visual Reasoning?" In: *CoRR* abs/1905.12506 (2019). arXiv: 1905.12506. URL: http://arxiv.org/abs/1905.12506.

[45] Kar-Han Tan and Boon Pang Lim. "The artificial intelligence renaissance: Deep learning and the road to human-level machine intelligence". In: *APSIPA Transactions on Signal and Information Processing* 7 (2018).

[46] Roman Vershynin. *Memory capacity of neural networks with threshold and ReLU activations*. 2020. arXiv: 2001.06938 [cs.LG].

[47] C. Wah et al. "The Caltech-UCSD Birds-200-2011 Dataset". In: CNS-TR-2011-001 (2011).

[48] Wei Wang et al. "A Survey of Zero-Shot Learning: Settings, Methods, and Applications". In: *ACM Transactions on Intelligent Systems and Technology* 10.2 (Feb. 28, 2019), pp. 1–37. ISSN: 2157-6904, 2157-6912. DOI: 10.1145/3293318. URL: https://dl.acm.org/doi/10.1145/3293318 (visited on 06/30/2020).

[49] Xiaolong Wang, Yufei Ye, and Abhinav Gupta. "Zero-shot Recognition via Semantic Embeddings and Knowledge Graphs". In: *CoRR* abs/1803.08035 (2018). arXiv: 1803.08035. URL: http://arxiv.org/abs/1803.08035.

[50] Yongqin Xian, Bernt Schiele, and Zeynep Akata. "Zero-Shot Learning — The Good, the Bad and the Ugly". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, HI: IEEE, July 2017, pp. 3077–3086. ISBN: 978-1-5386-0457-1. DOI: 10.1109/CVPR.2017.328. URL: http://ieeexplore.ieee.org/document/8099811/ (visited on 07/02/2020).

[51] Kuan-Ting Yu et al. "A Summary of Team MIT's Approach to the Amazon Picking Challenge 2015". In: *arXiv:1604.03639 [cs]* (Apr. 12, 2016). arXiv: 1604.03639. URL: http://arxiv.org/abs/1604.03639 (visited on 07/03/2019).

[52] Andy Zeng et al. "Robotic Pick-and-Place of Novel Objects in Clutter with Multi-Affordance Grasping and Cross-Domain Image Matching". In: *CoRR* abs/1710.01330 (2017). arXiv: 1710.01330. URL: http://arxiv.org/abs/1710.01330.

[53] Hao Zhang et al. "DoraPicker: An Autonomous Picking System for General Objects". In: *arXiv:1603.06317 [cs]* (Mar. 20, 2016). arXiv: 1603.06317. URL: http://arxiv.org/abs/1603.06317 (visited on 07/26/2019).