INVESTIGATING THE INFLUENCE OF COLOUR SPACES ON CONVOLUTIONAL NEURAL NETWORKS IN OPEN-ENDED 3D OBJECT RECOGNITION

Bachelor's Project Thesis

Anne-Jan Mein, s3399834, a.m.mein@student.rug.nl,
Supervisors: Dr. Hamidreza Kasaei

Abstract: Due to the rising use of service robots there has been an increased interest in high performing 3D object recognition architectures for open-ended environments. These architectures have successfully been created with the use of Convolutional Neural Networks utilising an open-ended learning approach. Adding colour-information to the object representation created by the architecture has demonstrated to increase performance. This study aims to examine the influence of colour-spaces on neural networks with regards to open-ended object recognition. This has been done by converting the colour-information of the object representation to the following colour-spaces: RGB, LAB, HSV, XYZ and YUV, as well as grayscaled images. These representations were then given as input for the state-of-the-art image classification networks: MobileNetV2, vgg16_fc1 and ResNet50. Three rounds of experiments were performed with the first two rounds utilising an offline- and last round an online evaluation. The first round to determine the best hyperparameter configuration for each network The second round to compare the colour-spaces, resulting in each network using a different colour-space to reach their highest performance. The online evaluation showed that vgg16_fc1 combined with the YUV colour-space achieved the highest object recognition performance in an open-ended setup. These results indicate that choosing the correct colour-space for an object recognition architecture utilising a Convolutional Neural Network can lead to a performance increase with regards to open-ended object recognition.

1 Introduction

The use of autonomous robots has become more and more widespread in recent years due to the improvement of artificial intelligence technologies, which in turn has lead to the increase in use of service robots (Wirtz et al., 2018). These service robots have to navigate highly dynamic environments where they are expected to perform, for example, grasping tasks with the use of object recognition.

Convolutional neural network approaches are often used for robotic vision as they have yielded good results in object recognition tasks (Wang et al., 2019). Multiple neural networks have been designed over the years with the goal of achieving high performance in object recognition tasks. However, dynamic environments are a challenge for these approaches, as it is impossible to completely train a network to completely prepare a robot for these environments (Yuille and Liu, 2021). To try and solve this issue, open-ended learning is utilised to create a high performing 3D object recognition architecture (Kasaei et al., 2015). This learning approach gradually introduces never-before-seen objects to the architecture which eliminates the need for extensive pre-training when used in combination with convolutional neural networks. This approach leads to fast object recognition which would allow service robots to make quick decisions in a dynamic environment similar to humans.

In order to further mimic humanlike behaviour studies have also looked at other aspects of robotic vision. Research shows that humans do not only utilise the shape, but also the objects’ colour when presented with a classification task (Bramão et al., 2011). Research done by Gowda and Yuan (2018) has shown that adding the colour-information of an object to an architecture can also improve the performance of object recognition tasks. This research also showed that transforming this colour-information to different colour-spaces leads to different classification performances in the same architecture.

The focus of this paper will be on the subject of encoding colour information in different colour-spaces and investigating the importance of colour information on object recognition in service robots. This will be combined with looking at multiple state-of-the-art convolutional neural networks to
examine the following research question:

- "In what ways do different colour-spaces influence neural networks with regards to open-ended object recognition?"

This is done by first looking at relevant related works in Section 2. Next, the proposed architecture is outlined in Section 3. A short explanation of the colour-spaces that are examined in this research is also presented as well as an overview of the two evaluation methods used to measure the performance of the architecture. The results of these evaluation methods are discussed in Section 4 and its implications will be concluded in Section 5 combined with proposals for future work.

2 Related Works

Service robots are expected to work autonomously in human-centric environments where fast object recognition is an important functionality due to the dynamic nature of such domains. This is often done with the use of three dimensional (3D) data, as it is more robust compared to two dimensional (2D) data due to less interference by factors such as illuminations and shadows (Regazzoni et al., 2014). To make sure that the proper information about a 3D object is extracted there needs to be a well-performing object descriptor. A recent study done by Kasaei et al. (2016b) introduced GOOD, a Global Orthographic Object Descriptor. The GOOD descriptor has a higher performance than other state-of-the-art descriptors, e.g., ESF (Wohlkinger and Vincze, 2011) and VFH (Rusu et al., 2010), with regards to overall classification performance. GOOD also had a better performance with regards to computational time, which was also a critique of ESF and VSH in a survey done by (Hana et al., 2018). This is an important critique to take into account what object descriptor to use for an open-ended scenario, as quick recognition is preferred.

Because humans use not only the shape, but also the colour of an object for recognition (Tanaka and Presnell, 1999; Bramão et al., 2011, 2012), there has been research that adds colour information to improve object recognition for robots (Gowda and Yuan, 2018). This is because, when colour information is ignored, different objects can look very similar (Ayoobi et al., 2020). This colour information can be added with the use of different colour spaces other than just RGB, leading to different performances (Gowda and Yuan, 2018). This has also been researched with the use of the GOOD descriptor to highlight increased performance when compared to using only the shape and texture information of an object (Ayoobi et al., 2020). This research indicates that colour-information is also valuable for robotic object recognition and should be taken into account when creating a robotic vision architecture.

Convolutional Neural Networks have been a very popular option for performing object classification tasks (Wang et al., 2019). It is, however, difficult to completely train a convolutional neural network for an open-ended environment that service robots operate in. This is due to memory and computational-time limits that come with trying to completely train these networks for such scenarios. Adding colour-information here mitigates this issue, as fewer parameters are needed for a relatively high accuracy (Gowda and Yuan, 2018). Furthermore, due to the difficulty of completely pre-training a neural network, there has been an increase in open-ended learning architectures as well (Lesort et al., 2020) (Kasaei et al., 2015) (Lucas, 1995). These architectures have also been applied to the GOOD descriptor with successful results (Kasaei et al., 2016a). An open-ended approach introduces multiple categories for the architecture to classify, and starts with few samples to simulate real-life, open-ended environments that service robots would work in. This approach utilises relatively little data and reduces the high memory and computational times mentioned earlier.

Three popular state-of-the-art convolutional neural networks that are used for image classification tasks are: MobileNetV2, vgg16.fc1 and ResNet50. The networks are popular due to being more accurate in image classification task compared to other convolutional neural networks (Sharma et al., 2018). There has also been research conducted showing that different colour spaces can have impact on performance with regards to image classification that utilises these networks (Kasaei et al., 2021). These networks have also been tested in open-ended environments, yielding high performance and showing that these networks are well suited for such tasks (of Kasaei, 2020).

3 Methods

The 3D object recognition architecture consists of multiple components. These components are: object detection, object representation, object recognition and object classification. The object detection component is responsible for the detection of an object in a scene. This component is not further discussed in this paper, as the datasets used already have the objects isolated. The object representation component takes detected object and transforms it in a way that can be used later for learning and classification. This learning and classification is done by the object recognition and object classification components. The object recognition component stores instances of objects in the
perceptual memory when learning and the object classification component utilises these when trying to classify new objects. The following subsections discuss these components.

3.1 Object Representation

The following two sections explain how the representation of an object is created with the use of an object descriptor (3.1.1) and what colour-spaces are used to add colour-information to this representation (3.1.2).

3.1.1 Object Descriptor

The Global Orthographic Object Descriptor, or GOOD, is used as the object descriptor for the experiments (Kasaei et al., 2016b). This method starts with constructing a global object reference frame of an object. It utilises principal component analysis on the point cloud in order to find the eigenvectors \([v_1, v_2, v_3]\). A point cloud is the representation of an object as a set of points in a 3D-space. Each point is described by their 3D coordinates, \([x, y, z]\), as well as RGB information. Using the directions of these eigenvectors the \(X\), \(Y\) and \(Z\) axes are obtained. The acquired reference frame is then used to create three orthogonal projections. These three projections are from the top, front, as well as the right-side. This is because bottom, back, and left-side are their mirrors. The projections are then divided in \(n\) by \(n\) bins that are subsequently used to compute a normalized distribution matrix by counting the amount of points that fall into each bin. This matrix is then used to create a histogram for each projection. The projection with the most information is considered the one with the highest entropy and is used to create a colour-, as well as a depth-image of the object for later use. The creation from 3D object to global feature vector can be seen in Figure 3.1.

![Figure 3.1: Overview of the creation of a global feature vector from a 3D object.](image)

3.1.2 Colour-Spaces

As stated in the previous sections, a colour-image containing the colour-information of an object is obtained from the projection with the highest entropy. This colour-information is important as it improves performance of image classification tasks (Gowda and Yuan, 2018). In this study we look at the following colour spaces: RGB, LAB, HSV, XYZ and YUV, as well as grayscaled images. These colour spaces were selected based on the works of Gowda and Yuan (2018) and Kasaei et al. (2021). Grayscaled images were added to see how networks would handle object classification with relatively little colour-information. The colour-images from the datasets are, by default, in the RGB colour-space. These images are then converted using the OpenCV python library. An example of an object displayed using these different colour-images can be seen in Figure 3.2.

The first colour space is RGB. This colour-space consist of three channels, Red, Green and Blue. Each channel has a range of values of \([0, 255]\), which is then normalized to achieve a range of \([0, 1]\). The LAB colour-space also consists of three channels. The first channel, \(L\), is for perceptual lightness and has a range of \([0, 100]\). The channels \(A\) and \(B\) are for the colours red, green, blue and yellow and have a value range of \([-128, 127]\). The third colour-space, HSV, stands for hue (\(H\)), saturation (\(S\)) and value (\(V\)). This colour-space was developed as an alternative to the RGB-space, as it is a closer representation of how humans perceive colour. The \(H\) channel has a range of \([0, 360]\), \(S\) and \(V\) of \([0, 255]\). The XYZ is a colour-space that is closely related to the RGB colour-space. The \(Y\) channel stand for luminance. The \(Z\) channel is close to the blue channel from RGB and the \(X\) channel represents a mix of three nonnegative RGB curves. An image in the RGB colour-space can be transformed into the XYZ colour space using the following transformation:

\[
\begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix} = \begin{bmatrix}
0.489989 & 0.310008 & 0.2 \\
0.176962 & 0.81240 & 0.0010 \\
0 & 0.01 & 0.99
\end{bmatrix} \begin{bmatrix}
R \\
G \\
B
\end{bmatrix}
\]

The values for the RGB channels are scaled from \([0, 1]\), so the values for the \(X\), \(Y\), and \(Z\) channel are \([0, 0.999997], [0, 0.990226]\) and \([0, 1]\) respectively. YUV also has three channels, one for luminance (Y) and two for chrominance (U and V). This means that the \(Y\) channel determines the brightness of the colour, while the \(U\) and \(V\) channels determine the actual colour. The \(Y\) channel has a range of \([0, 255]\), the \(U\) and \(V\) channels of \([-128, 127]\). The final type of colour-image is a grayscaled image of the object. In a grayscaled image each pixel has a value between \([0, 1]\) to represent its intensity. Here 0 means completely black.
and 1 completely white.

Figure 3.2: Different colour-images of a teapot. From top left to bottom right: RGB, LAB, HSV, XYZ, YUV and grayscale.

### 3.2 Object Recognition

After obtaining an object representation, three feature vectors will be created. These feature vectors contain the geometrical properties, colour and depth/texture information of the object. This part is handled by a convolutional neural network. To obtain a feature vector containing the geometrical properties of the object, the representation of the GOOD descriptor is used. The three projections are fed to a network in order to obtain this feature vector. Next a colour-image of the object is fed into a network to obtain a feature vector containing the colour information of the object. Then the same is done, but instead of a colour-image, the network gets fed a depth-image. The resulting feature vector contains the depth information of the object. The three obtained feature vectors, containing shape-, colour- and depth-information of a given object, are then concatenated to obtain one global feature vector. This concatenation is done using a pooling function or just appending the vector. These pooling functions are Max and Average pooling. This vector is then stored in the perceptual memory and can then be used later for learning and classification.

Three different state of the art convolutional neural networks are tested in this study. The three networks are: MobileNetV2, vgg16_fc1 and ResNet50. The configurations of these networks are the same as they are in their original research (Sandler et al., 2018) (Simonyan and Zisserman, 2014) (He et al., 2016).

### 3.3 Object Classification

Once the concatenated global feature vector is obtained, object classification is performed. This is done by comparing it to the feature vector of learned instances that are stored in the perceptual memory in order to calculate the dissimilarities. This is done by the use of the K-Nearest Neighbour algorithm combined with multiple distance functions. This approach is the same as the approach used by Kasaei et al. (2021), that utilises distances functions highlighted by Cha (2007). This is because the performance of the algorithm can change based on the distance function used. This is visualised in Figure 3.3, where K-Nearest Neighbour was used with different distance functions to classify points in a dataset. The following distance functions are compared: Bhattacharayya, Canberra, ChiSquared, Cosine, Dice, Divergence, Euclidean, Gower, Intersection, KLDivergence, Manhattan, Motyka, Neyman, Pearson, Sorensen and SymmetricKl. For a look at the mathematical equations the reader is referred to Cha (2007).

Figure 3.3: Results of point classification using different distance functions with the same k value of 5. A point falling into a colour indicates it is classified with said colour.

### 4 Experimental Results

Two different experiments are performed. The first is an offline evaluation, where the best setup for each of the three networks is obtained. The second experiment is an open-ended online evaluation.

#### 4.1 Offline Evaluation

The offline evaluation utilises the Restaurant RGB-D Object Dataset. This is a dataset containing 10 classes of objects, which can be observed in Figure 4.1. The dataset is relatively small, but has enough intra-class variation to make it suitable for performing extensive sets of experiments.

The evaluation utilises the same approach suggested by Kasaei et al. (2015), which is a k-fold cross-validation approach, to get the best configuration for each network. This means that the dataset is split into k groups, every iteration one is used for testing and the rest for training.

To determine the best configuration for each network, all possible configurations of bins, distance functions, pooling function and k for the
Class Accuracy is the same, then the configuration evaluated. The amount of bins is changed from 50 to 200 with increments of 50. There are a total of 3 different pooling methods and 15 distance functions, as was explained in Section 3.3, as well as a value of 1 through 9 for k, incremented by 2. This results in a total of \(4 \times 3 \times 15 \times 5 = 900\) experiments per network. These experiments were run with the use of RGB colour-images by default, as this is the colour-space that the objects from the datasets are in. Once the best configuration for each network is obtained we evaluate the performance of said configuration again, but with the 6 colour configurations detailed in Section 3.1.2. After these experiments have concluded we will have the best configuration of parameters for each network, as well as the colour-space that yields the best performance for that configuration.

The performance of each experiment is evaluated using three different criteria, Instance Accuracy \(\text{acc}_\text{micro} = \frac{\sum \text{true predictions}}{\sum \text{predictions}}\), Average Class Accuracy \(\text{acc}_\text{macro} = \frac{1}{K} \sum_{i=1}^{K} \text{acc}_i\), and Running Time. Note that we report average class accuracy to address class imbalance, since instance accuracy is sensitive to class imbalance.

The most important criteria is Instance Accuracy. The next criteria is the Average Class Accuracy and the final criteria is the Running Time of the experiment. The reason that three criteria are used is to determine an optimal configuration in the possible case that, for two or more configurations, their Instance Accuracy is the same. If this is the case then the best performing configuration is the one with the highest Average Class Accuracy. If both the Instance Accuracy as well as the Average Class Accuracy is the same, then the configuration with the shortest Running Time is chosen. This is because a fast configuration is preferred in an open-ended scenario over one that is slower.

4.1.1 Parameter Search

The results of the parameter search can be viewed in Tables 4.1, 4.2 and 4.3, where the top three configurations for each network can be observed. These results show that the networks perform best in the offline evaluation using different configurations. MobileNetV2 performs best with a resolution of 200 bins, while vgg_fc1 and ResNet50 perform best with 150 and 100 respectively. Both MobileNetV2 and vgg_fc1 perform best with MAX pooling, while ResNet50 has better performance with AVG pooling. The value for k is different for each network, with MobileNetV2, vgg_fc1 and ResNet50 using \(K = 9, K = 5\) and \(K = 1\) respectively. MobileNetV2 has the same Instance- and Average-Class Accuracy for all top three configurations, so the one with the lowest Running Time of 1.224 seconds was chosen. For vgg_fc1 the Instance Accuracy was the same for all three configurations, but because one has a higher Average-Class Accuracy of 0.947, that configuration was chosen as the most optimal. The same was the case for ResNet50, where the configuration with an Average-Class Accuracy of 0.963 was better than the other two.

Looking at the confusion matrices of these three configurations in Figure 4.3 it is possible to see which class was the most difficult for the networks to classify. The matrices show that each network had trouble correctly classifying object that belong to the Fork category and misclassifying them as a Spoon. This is likely because these object have a very similar size and shape in the used dataset.

4.1.2 Colour-Space Evaluation

The configurations obtained in the previous section were used to determine the best colour-space per network as was detailed in Sections 3.1.2 and 4.1. The results of these experiments can be seen in the Tables 4.4, 4.5 and 4.6 with the best colour-space highlighted in bold. Note that the Running Time is much longer due to this time including both the training and testing of the networks instead of only testing, which was the case with the experiments covered in Section 4.1.1.

![Figure 4.1: All ten object categories in the Restaurant RGB-D Object Dataset developed by Kasaei et al. (2015)](image-url)
Table 4.4: Colour-Space performance for the MobileNetV2 Network

<table>
<thead>
<tr>
<th>Colour-Space</th>
<th>Ins-Acc</th>
<th>Avg-Class-Acc</th>
<th>time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RGB</td>
<td>0.9739</td>
<td>0.9531</td>
<td>115</td>
</tr>
<tr>
<td>LAB</td>
<td>0.954</td>
<td>0.927</td>
<td>117.1</td>
</tr>
<tr>
<td>HSV</td>
<td>0.921</td>
<td>0.899</td>
<td>109.7</td>
</tr>
<tr>
<td>XYZ</td>
<td>0.967</td>
<td>0.942</td>
<td>119.5</td>
</tr>
<tr>
<td>YUV</td>
<td>0.958</td>
<td>0.930</td>
<td>117.2</td>
</tr>
<tr>
<td>Grayscale</td>
<td>0.958</td>
<td>0.927</td>
<td>168.9</td>
</tr>
</tbody>
</table>

Table 4.5: Colour-Space performance for the vgg16 fc1 Network

<table>
<thead>
<tr>
<th>Colour-Space</th>
<th>Ins-Acc</th>
<th>Avg-Class-Acc</th>
<th>time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RGB</td>
<td>0.967</td>
<td>0.947</td>
<td>260.2</td>
</tr>
<tr>
<td>LAB</td>
<td>0.967</td>
<td>0.949</td>
<td>257.4</td>
</tr>
<tr>
<td>HSV</td>
<td>0.928</td>
<td>0.902</td>
<td>268.5</td>
</tr>
<tr>
<td>XYZ</td>
<td>0.951</td>
<td>0.915</td>
<td>250.1</td>
</tr>
<tr>
<td>YUV</td>
<td>0.967</td>
<td>0.955</td>
<td>264.3</td>
</tr>
<tr>
<td>Grayscale</td>
<td>0.9642</td>
<td>0.9292</td>
<td>260.1</td>
</tr>
</tbody>
</table>

Table 4.6: Colour-Space performance for the ResNet50 Network

<table>
<thead>
<tr>
<th>Colour-Space</th>
<th>Ins-Acc</th>
<th>Avg-Class-Acc</th>
<th>time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RGB</td>
<td>0.974</td>
<td>0.963</td>
<td>186</td>
</tr>
<tr>
<td>LAB</td>
<td>0.958</td>
<td>0.959</td>
<td>177.1</td>
</tr>
<tr>
<td>HSV</td>
<td>0.938</td>
<td>0.911</td>
<td>180.4</td>
</tr>
<tr>
<td>XYZ</td>
<td>0.977</td>
<td>0.967</td>
<td>192.3</td>
</tr>
<tr>
<td>YUV</td>
<td>0.958</td>
<td>0.959</td>
<td>189.8</td>
</tr>
<tr>
<td>Grayscale</td>
<td>0.977</td>
<td>0.967</td>
<td>155.6</td>
</tr>
</tbody>
</table>

The results show that each of the networks has the best performance when the colour-images it uses for object learning and object classification are altered to a different colour-space. For MobileNetV2 this is the RGB colour-space which results in the highest Instance- and Average-Class Accuracy of 0.974 and 0.953 percent respectively. For vgg16 fc1 the best colour-space is YUV, as it has a slightly higher Average-Class Accuracy of 0.955 percent compared to RGB and LAB colour-space. Finally ResNet50 has the highest performance utilising Grayscale images as it has a slightly shorter Running Time of 155.6 seconds compared to XYZ’s 192.3 seconds.

The Instance- and Average-Class Accuracy of all three networks with their best performing colour spaces are all very similar, which can be seen in Figure 4.2, with ResNet50 slightly outperforming MobileNetV2 and vgg fc1.

4.2 Online evaluation

These experiments evaluate the network architecture in an open-ended environment. Each of the three networks as well as their best performing configuration and colour-space are evaluated in these experiments. The method utilised is the one adopted by Kasaei et al. (2015), which is a protocol that simulates the interaction of a robot with a real environment. The protocol defines a simulated teacher, this teacher interacts with the robot by performing on of three actions. With the Teach action the simulated teacher introduces a new object category to the robot. The Ask asks the robot what the category of a given object is. Finally the Correct action gives corrective feedback in the case of misclassification.

The main structure consists of the teacher introducing a new object category to the robot using three randomly selected views of the object using the Teach action. Using these views the robot then creates a model for that specific category. The teacher then presents the robot with a new object view to test if it learned the category or not using the Ask action. If misclassification occurs, then the teacher gives corrective feedback using the Correct action. The robot then updates that category model with the incorrectly classified view. Utilising this protocol it is possible to create an environment for the robot to learn and recognize objects at the same time.

The robot is pre-trained on the Washington RGB-D dataset (Lai et al., 2011). This large dataset contains 51 categories with 250,000 views of 300 objects. A new category is introduced by the teacher when the recognition performance is higher than a threshold $\tau$. The value for $\tau$ is set to 0.67, this means that the object recognition accuracy of the robot is at least twice as high as its error rate. If this threshold is not reached after 100 iterations, then the experiment is aborted and it is assumed that the robot cannot learn any more categories. The experiment is also aborted when the robot has learned all the 51 categories that are available in the dataset. The performance of the robot is influenced by the order of the categories/views that are
presented to the robot, this is why the experiment is performed 10 times for each network and then averaged.

The performance of robot is determined by 5 different evaluation metrics also used by Kasaei et al. (2021) and Kasaei et al. (2018). The first metric is called QCI, which stands for the number of Question/Correction Iterations. This is an indicator of how long it took the robot to learn. The second metric is called NLC and stands for Number of Learned Categories, this is an indicator of how much the robot could learn. The next method is AIC and stands for Average stored Instances per Category, this is an indication of time and memory resources utilised for learning. GCA and APA stands for the Global Category Accuracy and Average Protocol Accuracy respectively. These are both and indicator for how well the robot learns.

The results of the online experiments can be seen in table 4.7. As was discussed in Section 4.2, the experiment was run 10 times and the results were averaged. The configurations used were the ones obtained in section 4.1, which results in the following setups: MobileNetV2-RGB, vgg16_fc1-YUV and ResNet50-Grayscale.

When looking at the QCI criteria, it shows that MobileNetV2 and vgg16_fc1 both perform similarly with a QCI of 1327 and 1328 respectively, with ResNet50 having a slightly higher QCI of 135.

The results of the NLC of each network show that all three managed to learn the maximum of 51 object categories. This mean that the maximum number of categories that a network could learn was not possible to evaluate. As a result it is not possible to determine which network performs best with regards to the NLC criteria.

The AIC of each network was plotted in boxplots for a clearer visible comparison and can be seen in Figure 4.4. When looking at these results we can see that the performance of vgg16_fc1 is higher than the other two networks with value of 5.761. This means that vgg16_fc1 is more time and memory efficient as it only needs 5.761 instances of an object per category for learning. The difference between this and MobileNetV2 is little, but both are more time and memory efficient compared to ResNet50.

The GCA was also plotted for visibility and can be seen in Figure 4.5. This is because it is another important performance criteria when evaluating how well the system learns. The results here are similar to the results seen with respect to the AIC of the networks. vgg16_fc1 outperforms the best with a GCA of 0.894, with MobileNetV2 having a

Table 4.7: Results of the Online experiments, averaged over 10 runs

<table>
<thead>
<tr>
<th>Network</th>
<th>QCI</th>
<th>NLC</th>
<th>AIC</th>
<th>GCA</th>
<th>APA</th>
</tr>
</thead>
<tbody>
<tr>
<td>MobileNetV2-RGB</td>
<td>1906</td>
<td>1490</td>
<td>1843</td>
<td>97.39%</td>
<td>4027</td>
</tr>
<tr>
<td>vgg16_fc1-YUV</td>
<td>1906</td>
<td>1490</td>
<td>1843</td>
<td>97.39%</td>
<td>4027</td>
</tr>
<tr>
<td>ResNet50-Grayscale</td>
<td>1806</td>
<td>1390</td>
<td>1793</td>
<td>97.39%</td>
<td>4027</td>
</tr>
</tbody>
</table>

Figure 4.3: Confusion matrices of the best performing configurations for each network. From left to right: MobileNetV2, vgg16_fc1 and ResNet50

Figure 4.4: Average Stored Instances per Category for each network
were transformed into the following different colour-spaces:

- RGB
- LAB
- HSV
- XYZ
- YUV
- YIQ
- HED

This was done by using three state-of-the-art convolutional neural networks for image classification tasks: MobileNetV2, vgg16,fc1 and ResNet50.

Using the GOOD descriptor proposed by Kasaei et al. (2016a) these networks were given three orthogonal projections, as well as one depth image and one colour-image as input. The depth- and colour-image was created from the projection with the highest entropy. With these three types of input three feature vectors were created that were subsequently concatenated into one feature vector for the use of learning and classification.

These networks were first tested utilising the offline evaluation method to find the best performing hyperparameter configuration for the 3D object recognition architecture. All possible configurations of bins, distance functions, pooling function and k for the K-Nearest Neighbour were evaluated using colour-images in the default RGB colour-space.

Next, after obtaining the best configuration of hyperparameters for each network, the colour-images were transformed into the following different colour-spaces: RGB, LAB, HSV, XYZ and YUV, as well as grayscaled images. The offline evaluation was then performed again to determine the performance of each of the three networks when utilising the transformed colour-images. The results of these experiments showed that each network had the best performance using a different colour-space. MobileNetV2 had the best performance utilising the RGB colour-space, vgg16,fc1 with YUV and ResNet50 with gray-scaled images.

In the online evaluation the networks were tested with the corresponding colour-spaces that were found in the offline evaluation. The results showed that vgg16,fc1 had the highest overall performance in an open-ended object recognition setup, vgg16,fc1 outperformed MobileNetV2 and ResNet50 with regards to Average stored Instances per Category. This is an important metric when evaluating the performance of and open-ended object recognition system as these metrics determine the overall time and memory resources utilised when learning. vgg16,fc1 also had a higher Global Category Accuracy than the other two networks, which indicates that it learns better. The number of Question/Correction iteration and the Average Protocol Accuracy of vgg,fc1 was nearly identical to that of MobileNetV2 and all three of the networks reached the maximum Number of Learned Categories, which was 51.

These results indicate that colour-information does impact the performance of neural networks in open-ended object recognition tasks, which is in line with previous research (Kasaei et al., 2021). Using different colour-spaces leads to multiple differences in, for example, computational time and overall learning performance. This means that, when developing an architecture, the colour-space that the colour-information of an objects representation is in should not be ignored. Furthermore, MobileNetV2, vgg16,fc1 and ResNet50 all had their highest performance with a different network. This means that it is also necessary to determine the best colour-space when designing an open-ended 3D object recognition architecture utilising neural networks for object recognition.

All three networks learned the maximum number of categories, which was 51 in the Washington RGB-D dataset. This means that it is not possible to determine the maximum number of categories each network could possibly learn. To determine this, future research could use a dataset consisting of more categories. This allows for a more accurate identification of the network with the highest performance. Future research could also compare other colour-spaces that were found to yield high performance by Gowda and Yuan (2018); Keunecke and Kasaei (2021) like HED, YCbCr and YIQ.
References


