ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest appreciation to my supervisor, Professor **Masayuki MUKUNOKI** for his patient guidance, valuable suggestions, enthusiastic encouragement and the continuous support throughout my research work. His precious instruction, guidance and eagerness on my thesis gives strength to work out in all my research work. I am really appreciated for giving his time generously to me. Without him, this thesis would not possible. I also would like to thank Professor **Kunihito YAMAMORI** and Professor **Thi Thi Zin**, for their assistance on my thesis work.

I would like to give my sincere gratitude to the President of University of Miyazaki-Professor **Tsuyomu IKENOUE**, the members of my thesis committee, the professors for their applicable lectures that are appropriated for the research, the student affairs and GSO (International Relations Centre) family for their help to me.

I would like to express my immense sense of gratitude to Professor **Pyke Tin** and Professor **Thi Thi Zin** for their warmly carefulness like a family while I am studying in University of Miyazaki. I would like to say thanks to Myanmar DDP students and Myanmar Miyazaki Family for their valuable help to me whenever I need a help during staying in Miyazaki.

I also would like to appreciate to all our laboratory members for their warmest welcome, generous hospitality and sincerely friendship to me. And I would like to say deeply thanks to them for their kindness and giving time for all needed helps to me.

I would like to say thanks to Wei Liu, et al. and ImageNet ILSVRC Dataset for their advanced deep learning model and necessary dataset which give a lot of help for my research work.

Furthermore, I would like to express my special thanks to all person in JASSO (Japan Student Service Organization) for their support to study in University of Miyazaki.

Lastly, I wish to appreciate to my beloved parents for their greatest kindness and endless love. I am heartily thankful to my parents, teachers and friends for their affection.

ABSTRACT

In this research, we propose combination of transparent object feature region to SSD model for eliminating the false detections from the transparent object detection research. The detection of transparent object such as glass in the image is recently popular in computer vision researches. Among the various tasks of detecting objects in images, it is not an easy task to detect the presence of transparent objects in the image. The detection of transparent objects is very difficult to perform using classical computer vision algorithms since the appearance of transparent objects dramatically depends on its background and illumination conditions. In addition to the popularity of transparent object detection, deep learning is also giving high performance in object detection tasks. In this research, we apply one of the Convolutional Neural Network called Single Shot MultiBox Detector (SSD) for transparent object detection task. When we detect transparent objects with the network trained with glass images, many false detections are included in the detection results. In order to eliminate these false detections, we propose transparent object feature region to be included during the training processes. These object feature regions are the unique regions that appear due to the transparent properties of the glass objects. We manually define these object feature regions on each transparent objects and then train together with the glass training data. By using the network trained with glass and glass-feature regions, the glass and glass-feature regions are detected from images. The glass region which contains at least one glass-feature region is detected as a transparent object and the glass region without any glass-feature region is eliminated as non-transparent object. The experimental results show that the combination of transparent object feature regions to the deep learning model can considerably reduce the false detections and give a good performance to detect transparent objects in images.

Keywords - Transparent object detection, Deep learning, Single Shot MultiBox Detector, False detections, Transparent object feature region.

CONTENTS

PAGE

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1. INTRODUCTION

Nowadays, the recognition of different kinds of objects is increasingly challenging the computer vision researchers. Among these challenges, the recognition of transparent objects has become a considerable problem in object recognition task. Transparent objects are very widely used in daily life and are existing in domestic environment along with other objects. In contrast to the detection of other opaque objects, transparent objects are hard to detect by regular image segmentation methods because these objects usually take the texture from their background and their appearances are similar to their surroundings. Therefore, to perform the detection of transparent object in images, the advantages of deep learning techniques are intended to apply in this research.

Previously, many of the computer vision researchers had performed the segmentation of transparent objects in images by considering many of the features related with the transparent object. Their segmentation task had based on many physical features such as the material properties of objects like colour similarity between the background behind the transparent objects and the glass covered region, texture distortion, blurring cues, highlights caused by reflectance, and so on [1]. Based on these properties, they defined the features of transparent objects and performed the detection of transparent objects. Therefore, just in extracting features of the transparent objects, many of the physical properties should be considered. And again, detecting the location of transparent object in the image was hard to give the exact location of the transparent object in the image. These previous works were only based on traditional machine learning for the purpose of segmenting transparent object regions in an image. Comparatively, in deep learning techniques, the useful representations or features are learned directly from the given training images.

Deep learning is one of the machine learning algorithms which is structured based on how a human's brain works. Like the human's brain, deep learning techniques employ artificial neural networks with multiple layers which learn the features directly from the input data without the need for manual feature extraction. In a deep learning architecture, the first layer outputs a representative feature of the original input data and the successive intermediate layers use the output produced by the previous layer as input. The intermediate layer again outputs a new representative feature and then feeds to a higher level layer. Each level layer transforms its input data into a slightly more abstract and composite representation

and then passes to the next layer. In this way, the network learns multiple levels of representations that correspond to different levels of abstraction. Based on these multiple levels of representations, the final layer performs the detection of the objects.

Deep learning techniques have been widely used in many areas such as computer vision, speech recognition, natural language processing, audio recognition, social network filtering, machine translation, bioinformatics, drug design and board game programs which give results comparable to and sometimes exceed the human performance [2]. For object detection using deep learning, models are usually trained by using a large amount of labelled data from popular image datasets such as PASCAL VOC [3], COCO [4] and ImageNet ILSVRC dataset [5]. These datasets are released for object detection challenges.

With deep learning methods, object detection is performed by predicting the location of the object along with the class of the object. For predicting the location of the object, multiple windows of fixed sizes are applied over the input image. Each of these sliding windows are then passed to an image classifier to predict the class of the object in that window. In order to detect objects of different scales and different sizes, windows with different aspect ratios are used for sliding over the image. By this way, the detection of objects in an image is resulted with the predicting score of the object class and 4 variables (xmin, ymin, xmax and ymax) which represents the location of the object.

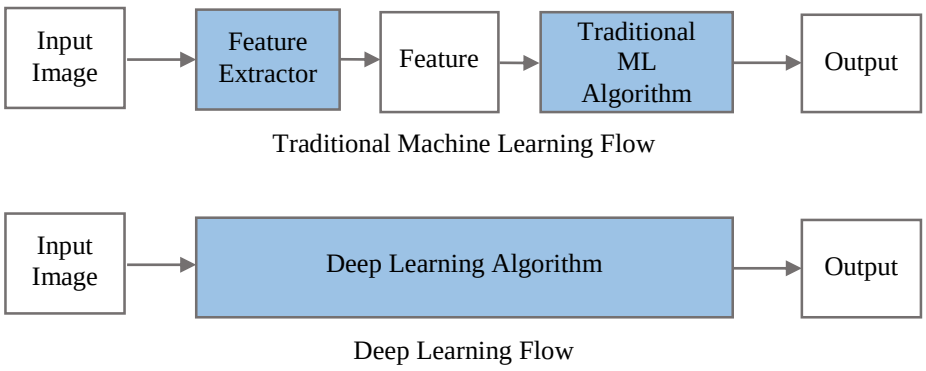


Figure 1. Machine Learning Object detection vs. Deep Learning Object Detection [6]

In this research, we use Convolutional Neural Network (CNN or ConvNet) for the detection of transparent object. With CNN, there is no need to do manual feature extraction which means the convolution layers of CNN extract the object features directly from the input image. Therefore, the features of transparent objects do not need to be considered during the

detection process. An illustration of machine learning object detection vs. deep learning object detection is shown in Figure 1. CNN learns to detect different features in layer-by-layer where the lowest layer starts with the simplest shapes such as edges of the transparent object and then simple shapes, complex shapes, more complex shapes and finally the shapes of the target transparent objects. By using the learned features of the transparent objects, a deep learning model that can detect transparent objects in images is created.

Among various deep learning models, we apply Single Shot Multibox Detector (SSD) [7] in this research. SSD is a state-of-the-art object detection model which uses a single net for both object localization and classification. SSD uses VGG-16 [12] with discarded fully connected layers as its base architecture. The architecture is added by a set of extra feature layers to be able to extract features of different scales and different sizes. Since SSD makes predictions for object detections based on feature maps produced at different stages of the convolutional neural network, it gives the accurate detection results in the detection of transparent objects.

Beyond the accurate detection of transparent object by SSD, one problem still exists when we detect non-transparent objects of the same shape as the transparent objects. If the detection is performed on such kind of non-transparent objects, SSD also classifies the non-transparent objects as the transparent objects. These false detection leads to a decrement in precision when detecting transparent objects from images. Although negative training examples of non-transparent objects can be used to solve the false detections, it is difficult to find several kinds of non-transparent objects which have the same shape as the transparent objects. Therefore, instead of trying to find each kind of non-transparent objects which have the same shape as the transparent objects, we need to find a way to concentrate only on the available transparent object data and also to be able to eliminate the false detection results as much as possible.

We propose a new transparent object detection approach which is based on the feature regions of transparent objects. We define some regions of the transparent objects as the important feature regions of transparent objects. These regions are the distinct regions which can appear only in the transparent objects due to their property of transparency. These regions are taken as the object feature regions and are included when training the neural network. The network then detects the transparent object regions and the object feature regions from

the input image. A glass region which contains at least one glass feature region is detected as a transparent object and the regions which do not contain any glass feature region are not detected as transparent objects. The experimental results show that the combination of transparent object feature regions to the convolutional neural network dramatically eliminates the false detections the transparent objects.

This thesis is composed of six chapters. This chapter provides an introduction of the research and a brief explanation of the objective of the research. Chapter 2 explains the theoretical background of deep learning and convolutional neural network. Chapter 3 includes related works to the detection of transparent objects, problem statements and then the proposed approach of the research. Chapter 4 gives details of our proposed method. Chapter 5 describes the conducted experiments, results and discussion on the research. Chapter 6 concludes this thesis.

CHAPTER 2. THEORETICAL BACKGROUND

2.1 Deep Learning and Object Detection

Deep learning is an advanced form of machine learning. In machine learning, the relevant features of objects need to be manually extracted at the beginning of the object detection process in order to create a model that can classify the object classes in the images. Therefore, we need to manually choose which features are relevant to the object that we want to detect and also need to choose the appropriate classifier for the object classification. With deep learning, a set of powerful algorithms are used to automatically extract the features to represent the data. In other words, the features that are relevant to the object that we want to detect are not pretrained and instead, these features are learned while training the network with a large set of training images. Feature extraction is done automatically in deep learning methods. That is why deep learning models are now giving a high accuracy and performance in many computer vision tasks such as visual object recognition and object detection.

When deep learning is applied in the object detection tasks, a large amount of labeled data need be prepared as training data. In order to perform automatic feature learning, deep learning uses the backpropagation algorithm which makes adaptation of the internal parameters of the deep neural networks to the training data. With a huge amount of training data, the internal parameters are changed more frequently so that the more accurate representation of the data are computed in each layer.

Another requirement when applying deep learning in object detection is the substantial computing power. Since deep learning uses huge amount of image data, it needs to perform thousands of matrix multiplications and other operations on images. The parallel architecture provided in high performance GPUs can make deep learning to be more efficient by parallelizing matrix and other processes and speed up the training processes.

The term “deep” in deep learning refers to the number of hidden layers which are included in the neural networks. Although the traditional neural networks have only two or three hidden layers, deep neural networks may contain tens or even hundreds of hidden layers in order to learn the representations of data with multiple levels of abstraction.

The most popular deep learning architectures are:

- Convolutional Neural Networks
- Deep Belief Networks
- Deep Auto-Encoders and
- Recurrent Neural Networks (or Long-Short Term Memory)

In this research, the deep learning architecture used for transparent object detection is the Convolutional Neural Network (CNN). The application of deep learning in object detection is shown in Figure 2.

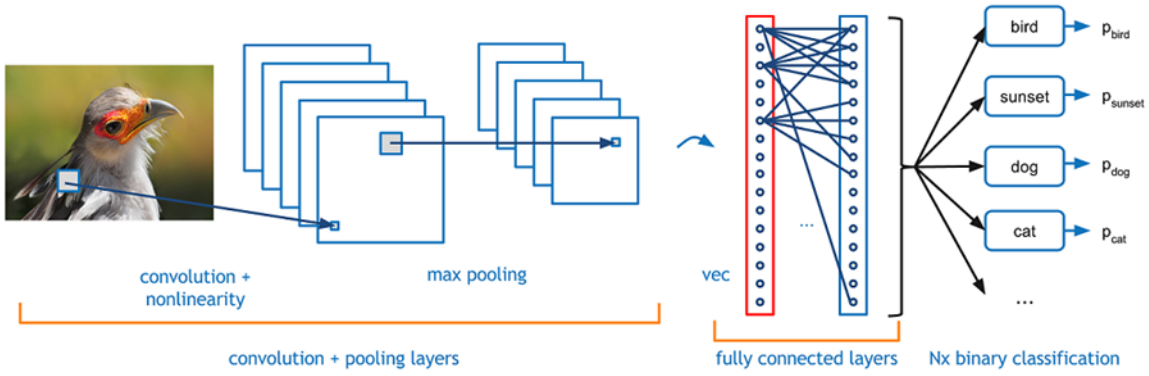


Figure 2. Object Detection using Deep Learning [8]

2.2 Transparent Object and its Characteristics

Transparent objects are the objects that allow the light to pass through them completely [9]. An example of transparent objects is the glass which can be easily found in our environment. Since the light can pass through the transparent objects, the objects inside or on the other side of these objects can be clearly seen by the human eyes. Generally, transparent objects are said to be colorless and they just only take the color of materials which are inside or behind them. Compared to transparent objects, non-transparent or opaque objects do not allow the light to pass through them and nothing can be seen through them.

The most obvious physical properties of objects are the color, texture and the reflectance of the light and these material features are usually used for describing different classes of the objects. For the detection of opaque objects, the color or texture property of the object is locally sufficient. However, transparent objects do not have their own color and texture because their appearance always varies according to the scene behind them. Therefore, some special properties have been defined for the transparent objects as below [1].

- Color similarity: the glass and background usually have the similar color due to the property of transparency.
- Highlights: specularities appear on the glass region because some lights are reflected from the glass surface.
- Blurring: the texture of the glass region is a blurred version of its background.
- Overlay consistency: the intensity distribution on a glass region is constrained by the intensity distribution on its background.
- Texture distortion: refraction in transparent objects causes a slight difference in the texture of the glass region than the others.

These properties had been used as the features for segmenting transparent objects in previous work [1]. In this research, we propose the transparent object feature regions caused by the property of transparency and train neural network with these feature regions for reducing the false detection of the transparent object in images.

2.3 Convolutional Neural Network

A CNN is composed by a sequence of convolution layers. Each layer filter or convolve the inputs to get useful features of the data. In order to extract the most important features of the data, the parameters of the convolution layers, also known as filters or kernel, are adjusted automatically by learning during the training process. A CNN architecture generally consists of three main types of layers: convolution layer, pooling layer and fully-connected (FC) Layer. Figure 3 shows a common architecture of convolutional neural network.

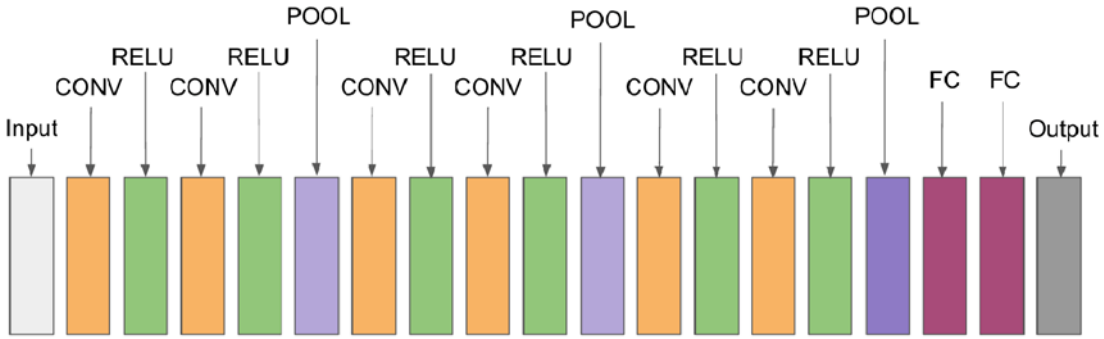


Figure 3. Common Architecture of Convolutional Neural Network [6]

The core of the convolutional neural network is the Conv layer and most of the computation are performed in these layers. The parameters of Conv layer are a set of learnable filters which are typically smaller width and height than the input image and have the same depth as the input image. As an example, a 5x5x3 filter (i.e. 5 pixels width, 5 pixels height and depth 3 for RGB colour channels) may be used in the first Conv layer. The filters are to slide or convolute over the entire input image and the dot product of the filter and the input is computed. During passing the filter, the visual features such as the edge of the object are learned. Starting from the lowest level feature such as edges, the Conv layers eventually learn the filters that activate the highest level feature such as the shape of the target object at the final layer of the network.

ReLU layers in the CNN architecture are the Rectified Linear Units. After computing linear operations such as element wise multiplications and summations at the Conv layers, a nonlinear function is applied at the ReLU layers. Although tanh and sigmoid functions were used as the nonlinear functions in the previous architectures, the CNN architecture mostly uses the function of $f(x) = \max(0, x)$ as the nonlinear function of the ReLU layer. According to this function, it can be said that ReLU layer changes all the negative activations to 0.

Pooling layers are another important layers in the CNN architectures. The purpose of the pooling layers is to perform non-linear downsampling. Among several non-linear functions, maximum pooling is the most common function used for pooling. With max pooling, the input image is firstly partitioned into a set of non-overlapping rectangles, and then the maximum value is output as the representation of each sub-region. After max pooling, the size of each representation is dramatically reduced and the amount of computation is also reduced. Therefore, pooling layers are important layers for controlling the overfitting caused by the Conv layers and are usually put between the successive Conv layers in the CNN architecture.

After several Conv and max pooling layers, the fully connected layers are inserted at the end of the CNN architecture. In the fully connected layers, the input is the activation maps of high level feature from the final Conv layers and the output is the possible classes and the probabilities of each class. Here, the possible classes are predicted by determining which high level features are the most strongly correlate to which particular class. For example, when a prediction is performed on an image of bird, there will be the highest activation in the wings or beak regions of the bird. Those features are particular for the bird so that the prediction will output the highest probability in the bird class.

The structure of the architecture varies according to the types of CNN architectures. The most popular convolutional neural network architectures are AlexNet [10], ZF Net [11], VGG Net [12], GoogLeNet [13], Microsoft ResNet [14] and R-CNNs [15]. Among these architectures, VGG Net, shown in Figure 4, is the CNN architecture used as the base network of the Single Shot Multibox Detector [7] deep model for detecting transparent objects.

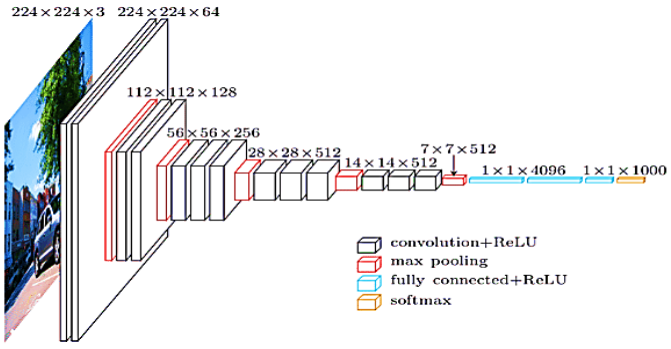


Figure 4. VGG-16 Architecture [16]

2.4 Single Shot MultiBox Detector (SSD)

Single Shot MultiBox Detector (SSD) [7] is a popular deep learning model which achieves not only faster detection result than the current state-of-the-art Faster R-CNN [17] but also higher accuracy than the YOLO [18] detection algorithm. SSD eliminates the proposal extraction stage and feature resampling stage from its architecture so that the detection is speeded up than the Faster R-CNN model. SSD makes predictions for detection of the objects based on features maps produced at different stages of the convolutional neural network. As a result, it can handle the detection of different scaled objects and produces a higher accurate detection result over the previous YOLO model. Therefore, SSD is providing a balance between speed and accuracy in the object detections.

SSD uses small convolution filters to predict object category scores and bounding box offsets and separate predictor (filters) to detect objects at different aspect ratios. Moreover, SSD can perform to achieve high detection accuracy by applying these filters to multiple feature maps and predicting at different scales. The main idea that SSD has introduced is to predict the object category scores and the location offsets for a fixed set of default bounding boxes by just applying small convolutional filters to feature maps. Due to its designated flow, SSD can perform end-to-end training and predictions of the object label and bonding boxes in a single pass. That is why it terms single shot. Since it also takes the feature maps from the layers which are closer to the original image, SSD can even perform the detection of objects in low resolution images. Therefore, SSD gives the accurate detection results on objects of different scales and different sizes and faster detection by single pass detection. The architecture of the SSD from [7] is given in Figure 5.

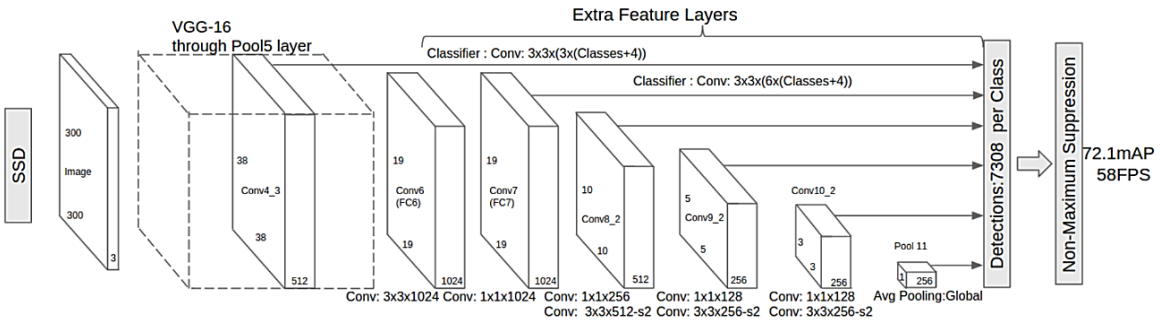
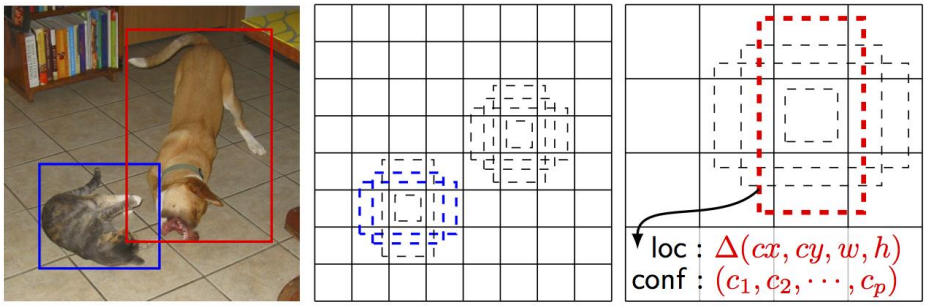


Figure 5. The Architecture of Single Shot Multibox Detector [7]

As shown in the above figure, SSD architecture uses VGG-16 with discarded fully connected layers as its base network. In place of discarded fully connected layers, a set of

extra feature layers (conv6 and later) are added in the architecture. Because of these added layers, it enables to extract features at multiple scales and resolutions.

During training, SSD only takes an input image and ground truth boxes of each object in the image. The input image is convoluted through layer-by-layer and output as several feature maps with different scales (e.g. 8x8 or 4x4 feature maps). With a larger feature map such as 8x8 feature map, small objects can be readily detected because the scale of each cell is smaller. With a smaller feature map such as 4x4 feature map, each cell covers a larger region of the image, thus enabling them to detect larger objects. On each scale of feature maps, a small set of default boxes (e.g. a set of 4 default boxes) with different aspect ratios (i.e. 2:1, 1:2 or 1:1) are slide at each location. Each box is defined with 4 offset values that is, the coordinates of the centre, the width and the height. For each default box, the prediction of these location offsets and the probabilities corresponding to the confidence over each class of object is performed simultaneously. These predicted boxes are matched to the ground truth boxes given at the start of training and then the best match box is labeled as positive and the others as negative. The framework of SSD [7] is shown in the following Figure 6.



(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

Figure 6. SSD Framework. [7] (a) Input image with ground truth bounding boxes of the cat and dog. (b) and (c) A small set of default boxes applied to feature maps of different scales (8x8 and 4x4). During training, the bounding boxes are matched to the ground truth boxes until we find the best match between them (blue boxes in (b) for cats and red box in (c) for dog).

2.4.1 Training Phase

Unlike the previous detectors that use region proposals, the ground truth information of the predicting object needs to be given since the training was started. After giving the images and ground truth box data, SSD applies the loss function and back propagation end-to-end to the network. During training, a lot of mechanisms are required to complete the whole training process. These mechanisms include choosing the suitable scales and aspect ratios of default boxes for performing scale variant detections, hard negative mining for excluding some default boxes and data augmentation for robustness to various sizes of the object in the input.

As the training objective, SSD is aimed to handle multiple categories of the object. Like the usual deep learning models, the goal of the training process is to find the parameter values that would optimize the training loss. Here, the overall training loss is the weighted sum of the localization loss (*loc*) and the confidence loss (*conf*):

$$L(x, c, l, g) = \frac{1}{N} \left(L_{conf}(x, c) + \alpha L_{loc}(x, l, g) \right) \quad (1)$$

where x is an indicator for matching default box and ground-truth box. c is the confidence over classes. l is the predicted bounding box and g is the ground-truth box. α is the weight term. N is the number of matched default boxes. If $N = 0$, the loss is set to 0.

The localization loss (*loc*) is the Smooth L1 loss which is measured by how far the bounding boxes predicted by the network (l) and the ground truth boxes (g). Since 4 parameters (*cx*, *cy*, *w*, *h*) compose to define the offsets of bounding box, the location loss of the network becomes:

$$L_{loc}(x, l, g) = \sum_{i \in Pos}^{N} \sum_{m \in \{cx, cy, w, h\}} x_{ij}^{k} \; smooth_{L1}(l_i^m - \hat{g}_j^m) \quad (2)$$

$$\hat{g}_j^{cx} = \left(g_j^{cx} - d_i^{cx} \right) / \, d_i^w \qquad \hat{g}_j^{cy} = \left(g_j^{cy} - d_i^{cy} \right) / \, d_i^h$$

$$\hat{g}_j^w = log\left(\frac{g_j^w}{d_i^w} \right) \qquad \hat{g}_j^h = log\left(\frac{g_j^h}{d_i^h} \right)$$

where *d* is the default bounding box, *i* means i^{th} position of bounding boxes and *j* means j^{th} ground-truth box.

The confidence loss (*conf*) is the softmax loss which is measured by how much the network has confidence (*c*) over the objectness of the predicted bounding box.

$$L_{conf}(x,c) = -\sum_{i\in Pos}^{N} x_{ij}^{p}\log(\hat{c}_i^p) - \sum_{i\in Neg}\log(\hat{c}_i^0) \qquad where \;\; \hat{c}_i^p = \frac{\exp(c_i^p)}{\sum_p \exp(c_i^p)} \quad (3)$$

and the weight term α is set to 1 in this thesis.

As one important mechanism, the default bounding boxes are carefully chosen to be at different dimensions and aspect ratios. Assuming *m* feature maps are used for detection, the scale of the default boxes is computed for each feature map by:

$$s_k = s_{min} + \frac{s_{max} - s_{min}}{m-1}(k-1), \qquad k \in [1,m] \quad (4)$$

where the value s_{min} is set to 0.2 for the lowest layer, s_{max} is set to 0.9 for the highest layer and the other inner layers are set to values that are periodically spaced.

The aspect ratios are then defined by a_r and set $a_r \in \{1, 2, 3, \frac{1}{2}, \frac{1}{3}\}$. After that, the size of each default box is computed by:

$$w_k^a = s_k\sqrt{a_r} \qquad and \qquad h_k^a = s_k\sqrt{a_r} \quad (5)$$

where *w* is the width of the default box and *h* is the height of the default box.

With chosen default bounding boxes, the prediction is made at each location of the feature maps. Each default boxes are matched against the ground truth boxes in order to select the most corresponding boxes to these ground truth boxes. Here, the matching strategy is to select the default boxes which have Intersection over Union (IoU) higher than a threshold 0.5 with any ground truth box. These selected boxes become the positive predictions and the other boxes becomes the negative predictions.

When selecting the bounding boxes with IoU higher than 0.5, the larger number of bounding boxes will have IoU lower than 0.5 and therefore will be denoted as negative predictions. A mechanism, called hard negative mining, is added to control the problem of larger amount of negative predictions over positive predictions. With hard negative mining,

the ratio of negative to positive examples is set to be 3:1 and only a smaller number of negative examples are used for training.

Data augmentation is another mechanism added to deal with various object sizes and shapes in the input. One option is to use each training image as its original image. The next option is to sample the original image with patches at different IoU ratios (e.g. 0.1, 0.3, 0.5 0.7, or 0.9) or random patches are used. By using one of these options, additional training data are generated so that the model becomes more robust to any size and shape of objects in the training images.

2.4.2 Detection Phase

The detection in SSD is different from the early deep learning models which use two different nets: Region Proposal Net (RPN) for extracting Region of Interest (RoI) and a separate classifier for object scoring and localization. Instead of using two separate nets, SSD uses a single net which predicts the object class and bounding box offsets simultaneously. In order to build up a single net for the whole detection process, extra feature layers are added at the end of the base network (VGG-16 with discarded fc layers). The base network is for high quality image classification and the added feature layers for predicting scale variant bounding boxes and their confidences.

There are three key elements used in the detection phase of SSD; multi-scale feature maps, convolutional predictors and default boxes of different aspect ratios. Once the image passes the convolution layers, they are resulted as the feature maps that represents the dominant features of the input image. The auxiliary feature layers added by SSD models decrease in size progressively, thereby producing feature maps that vary in scales and resolutions. By using multiple feature maps from different convolution layers, SSD promises the accurate detection on both of the large and small object in images.

After the feature maps output from the convolution layers, a fixed set of different convolution filters, called convolution predictors, are applied on different feature maps. When the filter is applied at each location of the feature map, it outputs the class score and the location offset related to the default box coordinates.

The default bounding boxes applied to feature maps are configured to come with different aspect ratios as mentioned in the training phase. Each default box is then applied at each feature map cell and predict the 4 offset parameters and object score of that default box region. If the number of default boxes used to predict c class scores and 4 location offsets is k, a m x n feature map will produce $(c + 4)kmn$ outputs.

Finally, Non-Maximum Suppression (NMS) mechanism is used to prune the bounding boxes which have a very low likelihood in prediction. A small confidence loss threshold (e.g. 0.01) is firstly used to filter out unlikely predicted boxes. Then the boxes with IoU lower than 0.45 are discarded from the remaining boxes. With non-maximum suppression, only the topmost predictions are retained by the network by eliminating the irrelevant predictions.

CHAPTER 3. TRANSPARENT OBJECT DETECTION AND FALSE DETECTION PROBLEM

3.1 Related Works

The detection of transparent objects had been a difficult work because of their lack of own appearance. The research on detection of transparent objects has been increasingly focused along with the development of intelligent domestic service robotics. In the domestic scenes, transparent objects are located among the other objects. For the detection of these transparent objects, Osadchy et al. [19] applied the specular highlights feature which makes glass objects different from the others. However, there was a requirement to have a light source. McHenry et al. [1] considered a number of features such as color similarity, blurring, overlay consistency and texture distortion in addition to highlights for transparent object detection purpose.

Fritz et al. [20] use an additive model of latent factors, method of a combination of SIFT and Latent Dirichlet Allocation (LDA) on a dataset of 4 transparent objects to generate transparent local patch appearance. The algorithm provides a useful result in the detection of transparent objects in different backgrounds.

Both of the detection and pose estimation of transparent objects have been proposed by Phillips et al. [21] and Lysenkov et al. [22] with the use of laser range finders and stereo and Kinect depth sensor, respectively. Phillips et al. [21] use inverse perspective mapping with the assumption of two views of a test scene and placing objects on a support plane. In [22], the fact that the Kinect sensor fails to estimate depth on specular or transparent surfaces is used to segment transparent objects from the images. And then, they perform 6 degree of freedom (6DOF) pose estimation and recognition of transparent objects. However, both of these approaches cannot handle overlapping transparent objects. So, Lysenkov et al. [23] propose an improved method to deal with the overlapped transparent objects.

As an interesting method of the segmentation of transparent object from a light-field image, Xu et al. [24] propose TransCut method using light field linearity, occlusion detector and graph-cut for pixel labeling. Unlike conventional methods which usually rely on the color

similarity and highlight information, they use the overlay consistency and texture distortion properties for the segmentation of transparent object region in a light-field image.

In recent years, the traditional object recognition tasks have been shifted to the deep learning object recognition tasks. Along with the powerful and efficient results, deep neural network is also applied to recognize transparent objects. Lai et al. [25] use Region with Convolutional Neural Network (R-CNN) to recognize the transparent object in color image. R-CNN technique uses selective search [26] to extract the interested region proposals [15], and the efficiency of the selective search algorithm is improved in [25] by considering the highlight and color similarity features of the transparent objects in order to remove some region proposals that are not transparent. As an interesting application of later deep neural network, we use Single Shot Multibox Detector (SSD) [7] to detect transparent objects in images.

3.2 False Detection Problem

Basically, SSD can perform the detection of all transparent objects accurately. But, one problem found in SSD is that if the detection is performed on the non-transparent objects of the same shape as the transparent objects, SSD also classifies the non-transparent objects as the transparent objects. Although the detection of transparent objects using SSD gives very accurate results on detecting transparent objects, some false detections are appeared in the results when non-transparent objects of similar shapes are detected. The false detections on two non-transparent objects of similar shape to transparent objects are shown in Figure 7.

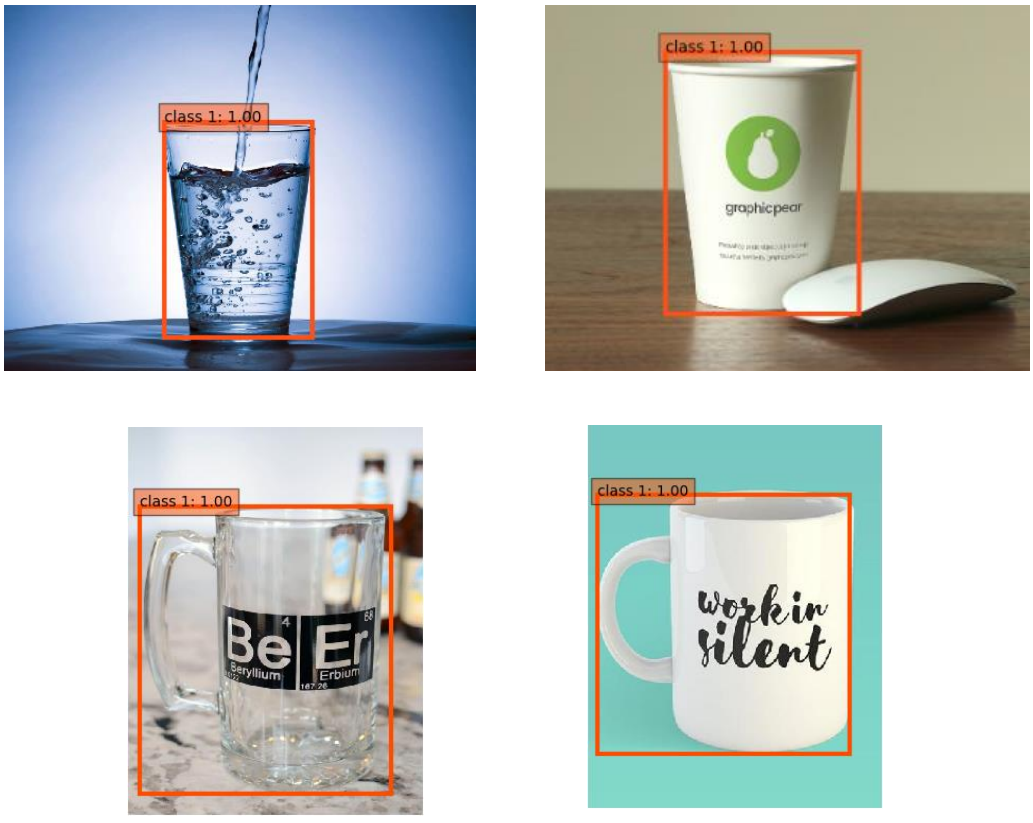


Figure 7. The false detection results of two non-transparent objects: paper cup with similar shape to water glass and coffee mug with similar shape to beer glass where class 1 is defined for glass class

This false detection problem appears because the appearance of the transparent objects is very simple and also they usually have no their own colour. In case of other objects, they have their unique features which are very distinct from the others. For instance, in case of bird, no other thing cannot have its distinct feature like wings. As for the transparent

objects, there is no special appearance and also the definite colour that the network can see during training.

There are two possible ways to solve this false detection problem. The first way is to train the network with the negative training examples which contains the non-transparent objects which have the similar shapes to transparent objects [27]. Although this is the regular way to solve false detection problem, it is very difficult to find several kinds of non-transparent objects which have the same shape as the transparent objects.

We propose an alternative way to eliminate the false detections in transparent object detection. In this approach, we absolutely do not find and add any kind of non-transparent objects of similar shape. Instead, the useful information is just only found from the available transparent object data for solving the false detection problem.

3.3 Transparent Object Feature Region for Eliminating False Detection

3.3.1 Transparency

Transparent objects are special objects come with different visual and physical feature compared to other regular object. One of its special properties is transparency [9]; the property that the rays of light passing through the glass medium so that the objects inside or beyond can be distinctly seen. Figure 8 visualizes the transparency property of some glass objects and compares to the same shape non-transparent objects which do not have transparency.

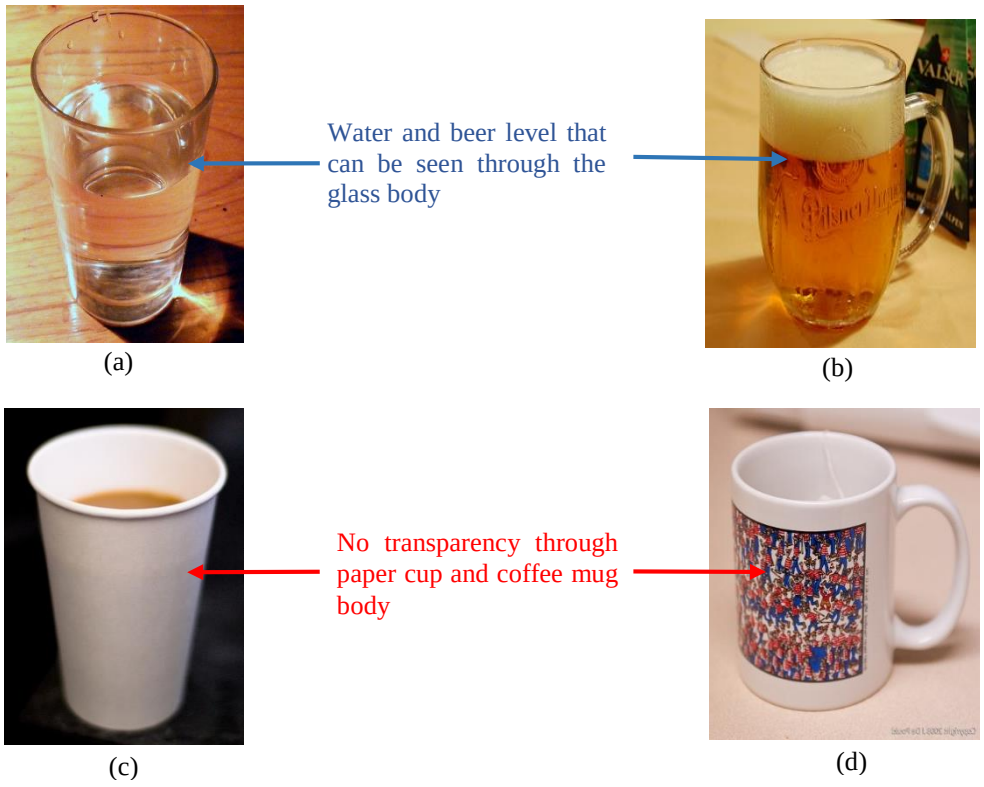
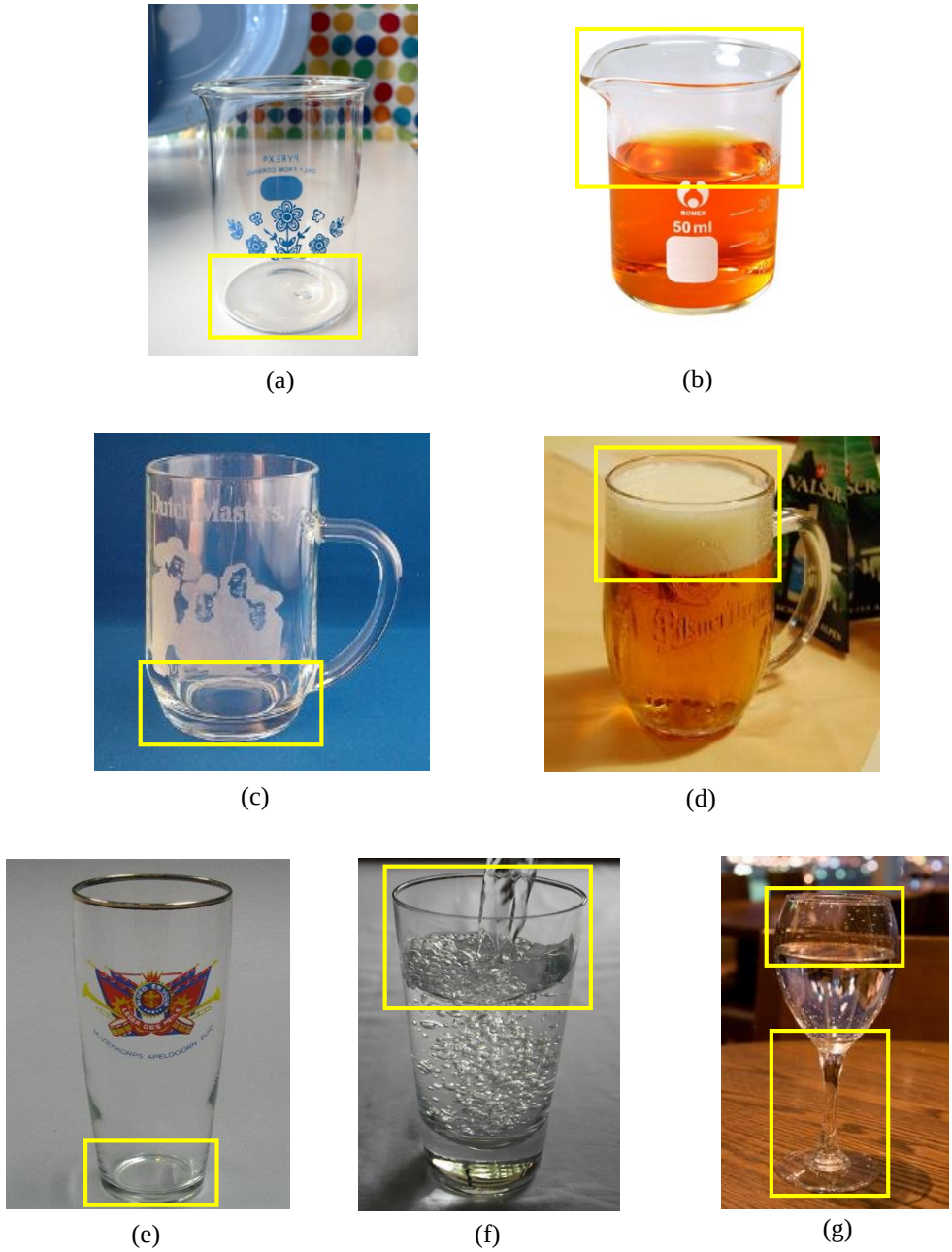


Figure 8. Transparency and non-transparency in glass and non-glass objects

Due to transparency in glass objects, the water and beer level can visually be seen through the glass body as shown in Figure 8 (a) and (b). Whereas, the contents inside the paper cup (c) and coffee mug (d) cannot be seen through because these non-glass objects do not have the transparency property. Based on this visual difference between them, we define the object feature regions on each transparent object. These feature regions are then included during training the convolutional neural network and discard false detections in the transparent object detection.

3.3.2 Transparent Object Feature Region

Transparent feature regions proposed in this system are varied according to each type of transparent objects. These regions are the most distinct parts that appear due to the transparency in the glass objects. Four types of transparent objects are used to perform the detection of transparent objects in this thesis. They are beaker, beer glass, water glass and wine glass, and these glass objects are common to be found in our environment. The feature regions are defined differently on each glass class as shown in Figure 9.



(a) (b) (c) (d) (e) (f) (g)

Figure 9. Object Feature Regions in different transparent objects

As shown in Figure 9, the transparent object features are defined at different regions of the glass. If the glass is empty like (a), (c) and (e), the bottom of the glass can be seen and that region is defined as the object feature regions. If the glass is filled with water or some liquid like (b), (d), (f) and (g), the region above the liquid level is taken as the glass feature region. In the case of wine glass (g), the feature region is defined to be the leg of wine glass if it is not filled with any liquid.

We create at least one feature region on each of the transparent objects in the training images. Therefore, when training the network, not only the whole glass but also the glass-feature regions are learned from each transparent objects. Since these feature regions are unique to transparent objects so that they cannot be detected in the non-transparent object of the same shape. When the detection is performed, the regions that do not contain any glass-feature region are regarded as non-glass objects, thereby eliminating the falsely detected glass regions from the result.

CHAPTER 4. OBJECT FEATURE REGION

4.1 Transparent Object Detection Using SSD

In this research, we apply SSD [7] as the convolutional neural network for detection of transparent objects in images. SSD uses VGG-16 [12] network which is pre-trained on large scale ImageNet dataset [5]. Pre-trained models are the models which have been trained with the general objects of different classes. With this pre-trained network, transfer learning is conducted for detection of transparent object. The concept of transfer learning is that the pre-trained model which has been learned the weight from other datasets are used instead of training the network from scratch with random initialization. When the pre-trained model is trained with the transparent object images, the network becomes a specific model for detection of the transparent object. Here, the network pre-trained on the ImageNet data is used because both the pre-trained network and the proposed network are intended for object detection.

For the detection of transparent object, the training dataset is taken from the ImageNet ILSVRC dataset. The ILSVRC dataset provides some classes related with transparent objects and we use 4 classes:

- Beaker
- Beer glass
- Water glass and
- Wine glass

For each class, the images and annotation files are given. The annotation files are the files which are related to each image and describe the location of objects in the image along with their labels. For some class such as water glass, the annotation files are not readily given and we create them manually for using in training the neural network. An example of annotated image provided by the ILSVRC dataset is shown in Figure 10.

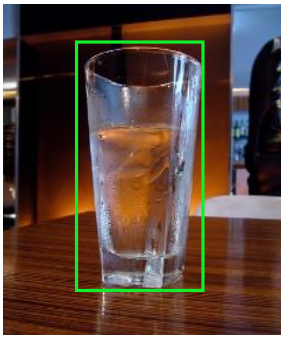


Figure 10. Image with annotated bounding box

We prepared a total of 1,568 images and their annotation files from beaker, beer glass and water glass classes for training the convolutional neural network. Since each image contains one or more transparent objects, the number of bounding boxes for the transparent objects is 1,888 boxes from all classes of the transparent objects.

The convolutional neural network is then trained with the images and bounding box annotation files provided by the ILSVRC dataset. Using the trained network, transparent objects are detected from the images. In the detection output of SSD, the position of the detected objects is shown with the bounding boxes and, for each bounding box, the class label and the score for the class are described. In this system, we define the class label 1 for the transparent objects. Since the score of the detection is the probability of how much the detected region has the same features as the transparent objects, it is described by the value between 0 and 1. Some of the detection results of transparent objects in images are shown in Figure 11.

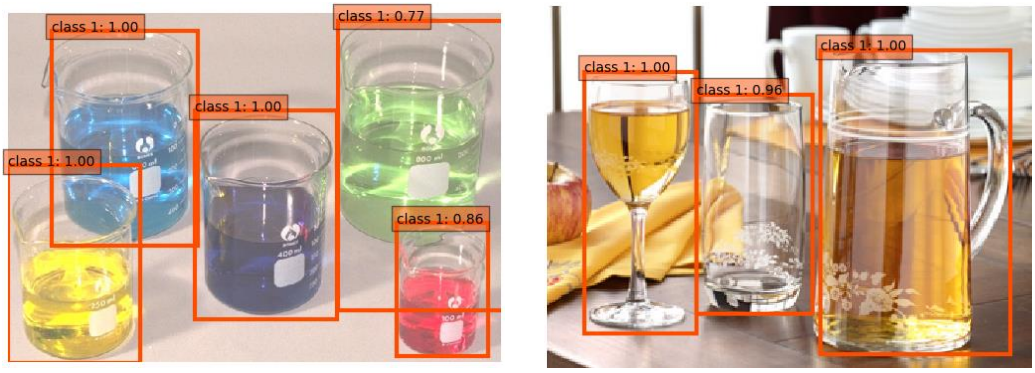
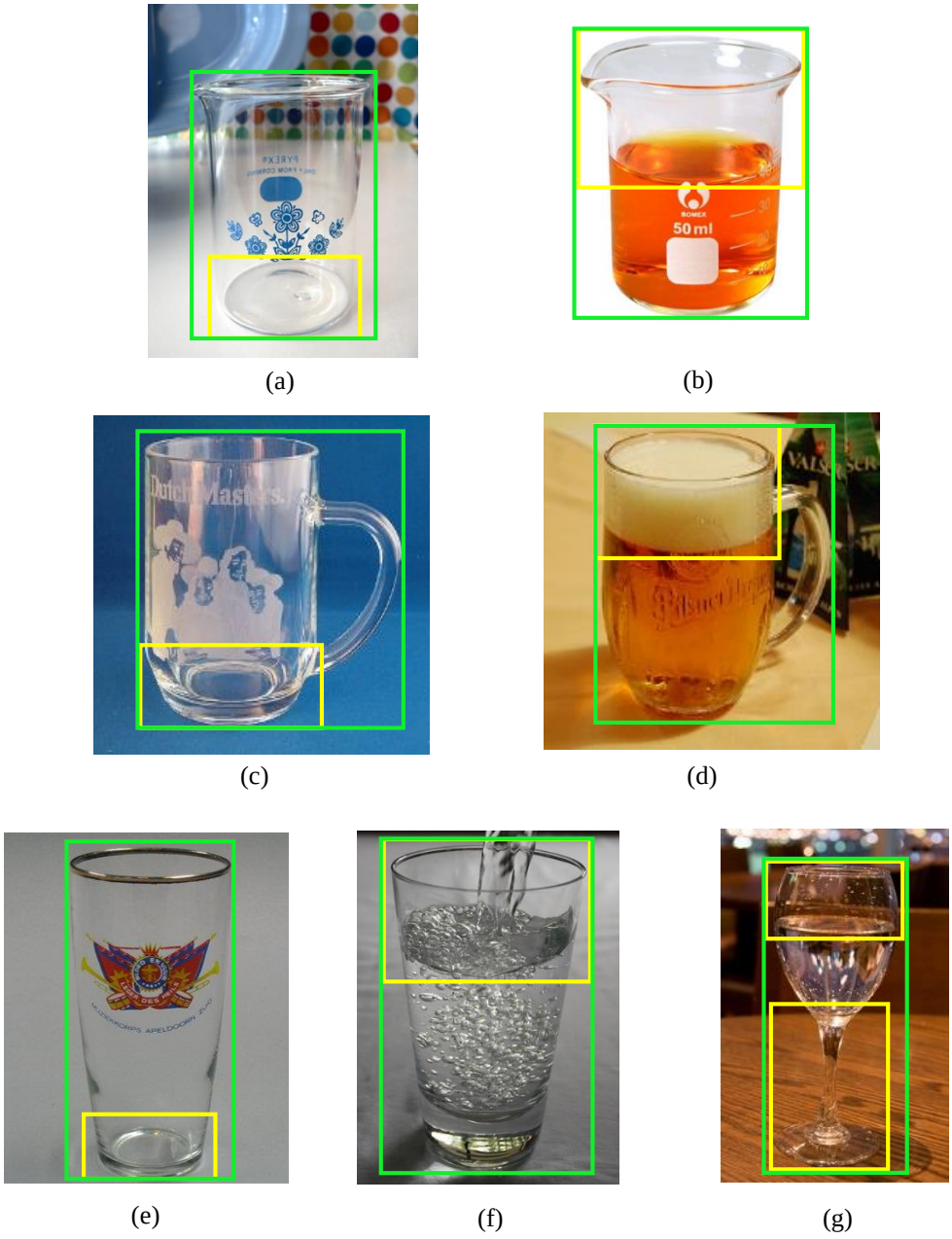


Figure 11. The detection results of transparent objects in images where class 1 represents the label of glass object

Although the detection of transparent objects from images which contain only the transparent objects, some of the false detections appear when the network is tested on the images which contain non-transparent objects of the same shape as transparent objects. These false detections can degrade the performance of transparent objects detection. Therefore, the transparent object feature region is proposed in this research and deal with the false detections.

4.2 Training with Transparent Object Feature Region

For our proposed object feature regions in transparent objects, we created additional bounding boxes on transparent objects as shown in Figure 12.



(a) (b) (c) (d) (e) (f) (g)

Figure 12. Training images annotated with both glass region (green box) and glass-feature region (yellow box) for each transparent object class: beaker, beer glass, water glass, and wine glass. Some glass-feature regions are defined at the bottom side of the glass object (a), (c), (e) and lower yellow box in (g). Some glass-feature regions are defined at the top side of the glass object (b), (d), (f) and upper yellow box in (g).

When training the network, different labels are used: class 1 for glass and class 2 for glass-feature regions. After training the network with the proposed data, the network is tested on the same images and the results are output as shown in Figure 13.

(a) (b)

(c) (d)

Figure 13. The detection result after training the network with glass and glass-feature region data. The label class 1 represents the glass and class 2 represents the glass-feature.

As can be seen in the final detection results, the glass-feature regions are only detected in the transparent objects and they are not detected in the non-transparent objects. This is the expected output that we propose to eliminate the false detections from the detection results.

The glass regions which do not contain any glass region are regarded as the non-transparent objects and eliminate these regions from the detection results. Therefore, the paper cup in Figure 13 (b) and the coffee mug in (d) are not detected as the transparent object because they do not contain any glass-feature region. By this way, the false detections are dramatically decreased in the detection results and the precision in the detection of transparent objects has been increased.

CHAPTER 5. EXPERIMENTATION AND PERFORMANCE EVALUATION

5.1 Performance Evaluation

5.1.1 Training Data

For training the network, the dataset is taken from ImageNet ILSVRC dataset [5]. Since ImageNet dataset does not provide annotation files for some training images, we manually created the annotation files for training the network. The number of images and annotated bounding boxes used in our experiments are shown in Table 1.

Table 1. The number of annotated bounding boxes in each class of training images.

Object Classes		Num. of Images	Num. of object bounding boxes	Num. of glass-feature bounding boxes
Transparent Objects	Beaker	411	541	541
	Beer glass	345	379	407
	Water glass	282	302	319
	Wine glass	530	666	666
Total		1,568	1,888	1,933
Non-transparent Objects	Paper cup	500	648	-
	Coffee mug	200	230	-
	Coffee cup	200	223	-
Total		900	1,101	-
Negative training objects	Bicycle	138	158	-
	Car	150	168	-
	Airplane	150	168	-
	Child	85	94	-
	Cat	163	168	-
	Dog	150	162	-
	Table	150	156	-
	Chair	150	183	-
	Clock	150	153	-
Total		1,286	1,410	-

In this experimnets, we use the programs provided at [28]. During training, we use VGG_ILSVRC_layers_fc_reduced.caffemodel [29] and train the network at a maximum iteration of 120,000. Other training parameters are set to default values.

5.1.2 Testing Data

For performance evaluation of the detection results, a set of 400 testing images which contains 202 images of transparent object and 198 images of non-transparent object are used. All of these testing images are images which are not used during training the network. For each of these test images, the ground-truth bounding boxes, in other words, the true locations of the objects are created to compared with the bounding boxes predicted by the trained network. An example of ground-truth bounding box and predicted bounding box is shown in Figure 14.

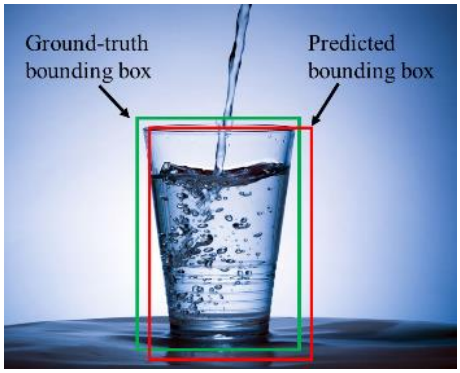


Figure 14. Ground-truth bounding box and predicted bounding box over the detected object

The number of images and ground-truth bounding boxes in each class of the testing images are described in Table. 2.

Table 2. The number of ground-truth bounding boxes in each class of testing images.

Object Classes		Num. of Images	Num. of annotated bounding boxes
Transparent Objects	Beaker	45	109
	Beer glass	52	107
	Water glass	66	97
	Wine glass	39	94
Total		202	407
Non-transparent Objects	Paper cup	52	162
	Coffee mug	74	129
	Coffee cup	72	116
Total		198	407

5.1.3 Calculating TP and FP Using IoU

Using the ground-truth bounding boxes and the predicted bounding boxes, we evaluate how precisely the network can detect the transparent objects in images. Intersection over Union (IoU), also called Jaccard Overlap index [30], is calculated for each detection.

$$\text{IoU} = \frac{A \cap B\ (Area\ of\ Overlap)}{A \cup B\ (Area\ of\ Union)} \tag{6}$$

In the above equation, A represents the area of the ground-truth box and B represents the area of the bounding box predicted by the network. Then, the IoU of each detection is calculated by dividing the overlap area of A and B by the union of the area of A and B. The calculations of IoU between ground-truth bounding box and different predicted bounding boxes are shown in Figure 15.

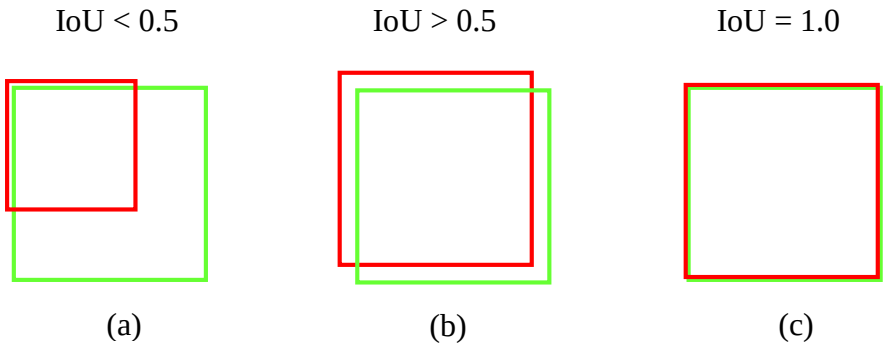


Figure 15. Different IoU calculated over the predicted bounding box and the ground-truth bounding box [28]: (a) Poor detection, (b) Good detection, and (c) Excellent detection

The higher IoU value means the more accurate the network can detect the object regions and the lower IoU value means the poor detection. The IoU threshold value of 0.5 is used in our experiments for precise detection of the transparent objects.

Then the true positive (TP) and false positive (FP) are calculated. If the IoU between the predicted bounding box and ground truth bounding box is greater than or equal to 0.5, the detected region is defined as a TP detection and if the IoU is lower than 0.5, it is assumed to be a FP detection.

5.1.4 Precision, Recall and F-measure

Precision is the measurement of how accurately the network can make predictions of the objects. In other word, it describes the percentage of the true detections over the whole detection result.

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

where, TP = Number of True Positive (Detection with IoU ≥ 0.5), FP = Number of False Positive (Detection with IoU < 0.5).

Recall is the measurement of how many true detections that the network can predict. In other word, it calculates the percentage of predicted true detections over the ground-truth detections.

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

where, FN = Number of False Negative (Not detected ground-truth)

The precision and recall are inversely related so that if the precision gets higher, there will be some decrement in recall and if the recall gets higher, there will be a decline in precision. Therefore, we need to balance the precision and recall by trying to get a network that finds only relevant objects and detects all ground-truth objects.

F-measure (also called F1 score or F-score) is the harmonic mean of the precision and recall. The value of F-measure becomes the highest (i.e., 1) if there is a best balance between precision and recall, and the value decreases as the balance between precision and recall is lower. F-measure is calculated by the following formula:

$$F\text{–}measure = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{9}$$

5.1.5 Average Precision (AP) and mean Average Precision (mAP)

Average Precision (AP) is different from the precision calculated by the ratio of true detections and total detections. For each detection, the network also outputs the predicting score for each possible class and these scores varies for each predicted bounding box. Moreover, there is also IoU value calculated on each prediction. The best detection result is the prediction with true class score 1.0 and IoU value 1.0.

For calculating AP, precision and recall are calculated at different class scores and plotted on a precision-recall curve. A good object detector results a precision-recall curve where the precision stays high as the recall increase. The recall is divided into *r* levels and the maximum precision values at each recall level, called interpolated precision p_{interp}, is calculated for any recall level $r' \geq r$ by:

$$p_{interp}(r) = \max_{r' \geq r} p(r') \tag{9}$$

Average Precision (AP) is computed by averaging the interpolated precision at the different recall levels. For instance, if the recall is divided with 11 levels and each level is separated by 0.1 interval, the AP is calculated by:

$$AP = \frac{1}{11} \sum_{r \in \{0.0,\ldots,1.0\}} p_{interp}(r) \tag{10}$$

Average Precision (AP) is calculated on each class of the object and the mean Average Precision (mAP) is calculated over all classes of the objects contained in the detection system. If the system detects *N* object classes, mAP is calculated by averaging the AP of all *N* object classes and defined by:

$$mAP = \frac{1}{N} \sum_{n \in N} AP(n) \tag{11}$$

5.2 Network Trained with Glass and Glass-feature

This time, the network is trained by our proposed method. Although the false predictions can be also reduced by training with non-glass data, these kinds of same shape data may not always be available for further classes of glass classes. This proposed method does not depend on the availability of the same shaped non-glass data. Instead, the network is trained with only the available glass images by giving the glass and glass-feature regions in these images. During training, the glass region is learned as class 1 and glass-feature region as class 2.

After the network is trained with glass and glass-feature, the network is also tested on testing dataset and produce the output images as shown in Figure 16.

Figure 16. Some detection outputs of the network trained with glass and non-glass

As can be seen in the detection results, some regions in the non-glass images are falsely detected as the glass-feature regions. Since these regions can make non-glass objects to be detected as glass object which contains the glass-feature region, these falsely detected glass-feature regions on non-glass objects are needed to be removed. Therefore, we set some

thresholds to glass-feature regions and the glass-feature regions whose confidence score less than the threshold are removed. The threshold values are from 0.0 to 0.7 and increase by 0.1. While increasing the threshold of glass-feature region, most of the glass-feature regions in the glass objects are not removed because they have a high confidence score on prediction as glass feature. But, the glass-feature regions in non-glass objects are predicted with low confidence score and, therefore, are removed when they become lower than the threshold. By this way, most of the falsely detected non-glass objects are eliminated and the network achieves a high performance in detecting transparent objects. Among different thresholds for the glass-feature region, threshold value of 0.3 gets the highest mAP results.

5.3 Comparison Methods

In this research, many kinds of experiment are conducted in order to compare and evaluate the performance with our proposed method. The network is trained with different training data and settings.

5.3.1 Network Trained with Only Glass

For the detection of transparent objects, we firstly train the network with glass images annotated only on the glass regions. We use this network for comparing with the proposed network because the network trained with only glass is the base line network for the detection of transparent objects. A total of 1,568 transparent object images and its annotation files are used to trained the network. In this training process, image background is trained as class label 0 and the glass region as class 1.

5.3.2 Network Trained with Glass and Augmented Glass Data

Data augmentation is a popular concept to increase the number of training data. With a larger training data, the network can learn more training samples so that this network could be one option for improving performance of the transparent objects detection. Therefore, we use this network trained with glass and additional augmented glass data to compare with our proposed method. The original glass training images are horizontally flipped and created as the augmented glass training images as shown in Figure 17.

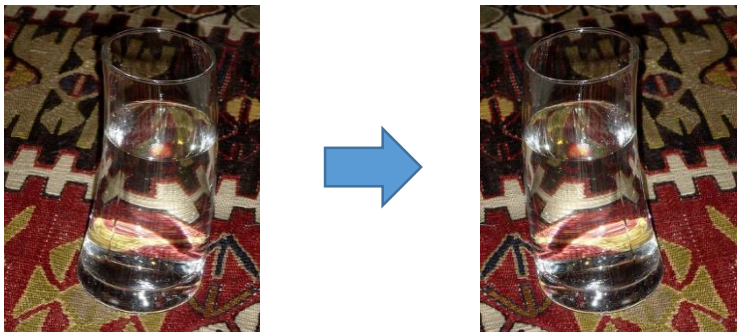


Figure 17. Original glass image and horizontal flipped glass image

The network is then trained with a total of 3,136 images: 1,568 original glass images and 1,568 augmented glass images.

5.3.3 Network Trained with Glass and Negative Training Data

To decrease the FP detections, we need to train the network to learn which are positive samples and which are negative samples. Since training the network with both positive and negative training data is another option to decrease false detections, we use this network for comparison.

The negative training samples are found from the available training data. In the ImageNet dataset, some object classes are readily available and are given along with annotation files. Images from these classes are tested on the network trained with only glass and find the falsely detected regions. These falsely detected regions are annotated as the negative object class and the network is trained with glass and these negative training samples. The object classes that are used for creating negative training samples are: bicycle, car, airplane, child, cat, dog, table, chair and clock. From these 9 classes, a total of 1,269 images are used as the negative training data. The network is therefore trained with 1,568 glass images and 1,269 negative training images (Table 1).

5.3.4 Network Trained with Glass and Non-glass

For reducing FP in detection results on non-transparent objects of the same shape, the most common method is to add those non-transparent objects in the training dataset. Therefore, we use this network as one comparison method of reducing false detections of the transparent objects.

Although this approach is useful, it sometimes has difficulty to match which non-glass objects have the same shape as the glass objects. In this research, we find some non-glass classes which have the most similar shape to glass objects and included in training the network. The non-glass classes are: paper cup, coffee mug and coffee cup and a total of 900 images from 3 classes are used as non-glass training examples (Table 1). Here, the glass classes are labels as glass and three non-glass classes are label as non-glass to train the network.

5.4 Performance Comparison of Different Training Processes

5.4.1 TP and FP Comparison

In this research, a total of 5 training processes are conducted and their performance are calculated on precision, recall and mAP matrices. In our experiments, the score threshold for glass class is set to 0.5. The calculated performance of these training processes are described with the following tables.

Table 3. The number of TP and FP in different training processes.

Network		TP	FP	Ground-truth
Network trained with only glass		394	550	407
Network trained with glass and augmented training data		**397**	**661**	407
Network trained with glass and negative training samples		395	505	407
Network trained with glass and non-glass	Glass	**397**	**175**	407
	Non-glass	377	172	407
Network trained with glass and glass-feature	Glass-feature th 0.0	397	374	407
	Glass-feature th 0.1	392	198	407
	Glass-feature th 0.2	392	164	407
	Glass-feature th 0.3	**390**	**143**	407
	Glass-feature th 0.4	387	135	407
	Glass-feature th 0.5	380	132	407
	Glass-feature th 0.6	371	120	407
	Glass-feature th 0.7	362	102	407

In Table 3, the number of TP and FP in different training processes are described. When the network is firstly trained with only glass data, its TP is 394 out of 407 ground truth but its FP is very larger than the TP. When the training data is increased with augmented glass data, both of the TP and FP have been increased. Then, the network trained with negative training data from 9 classes gives a result with a little bit higher TP and a less number of FP. But, the number of FP is still higher than the number of TP in these three processes. When the network is trained with non-glass of the same shape as glass objects, the FP are considerably decreased. But, this training process can have some limitation in the availability of training data in the long term. With the network trained with glass and glass-feature, the number of FP is reduced much lower than all other training processes without decreasing too much in the number of TP.

5.4.2 Precision and Recall Comparison

Table 4. Precision and recall calculation in different training processes.

Network		Precision $TP/(TP+FP)$	Recall $TP/(TP+FN)$	F - measure
Network trained with only glass		41.74 %	96.81 %	58.33 %
Network trained with glass and augmented training data		37.52 %	**97.54 %**	54.19 %
Network trained with glass and negative training samples		43.90 %	97.05 %	60.45 %
Network trained with glass and non-glass		69.05 %	95.09 %	80.00 %
Network trained with glass and glass-feature	Glass-feature th 0.0	51.49 %	97.54 %	67.40 %
	Glass-feature th 0.1	66.44 %	96.31 %	78.63 %
	Glass-feature th 0.2	70.50 %	96.31 %	81.41 %
	Glass-feature th 0.3	**73.17 %**	**95.82 %**	**82.98 %**
	Glass-feature th 0.4	74.13 %	95.09 %	**83.31 %**
	Glass-feature th 0.5	74.22 %	93.37 %	82.70 %
	Glass-feature th 0.6	75.56 %	91.16 %	82.63 %
	Glass-feature th 0.7	78.02 %	88.94 %	83.12 %

According to calculated data in Table 4, the recall of all training processes are not too much different and most of them achieve a recall rate of over 95.00 %. But, the data shows too much difference in precision rates. Although the network trained with glass and augmented glass data has the highest recall rate, its precision is too much lower than the other networks. The other two networks: network trained with only glass and the network trained with glass and negative training data achieve higher precision rates than the network trained with glass and augmented glass data. But, their precision rates are still lower than 50.0 % and may not be a reasonable precision rate. The precision rate becomes 69.05 % when the network is trained with glass and non-glass. In the case of network trained with glass and glass-feature data, the precision rates become the highest rates among all of the training processes. Even with the original available glass training data, the combination of transparent object feature in the glass detection achieves the higher precision and recall when the glass-feature threshold is set to 0.3. With a lower number of data used during training process, our proposed method shows the highest F-measure among all comparison methods.

5.4.3 Average Precision (AP) Comparison

The average precision (AP) are calculated based on precision-recall curve of the detection result and Figure 18 shows the precision-recall curves and AP results of different training processes.

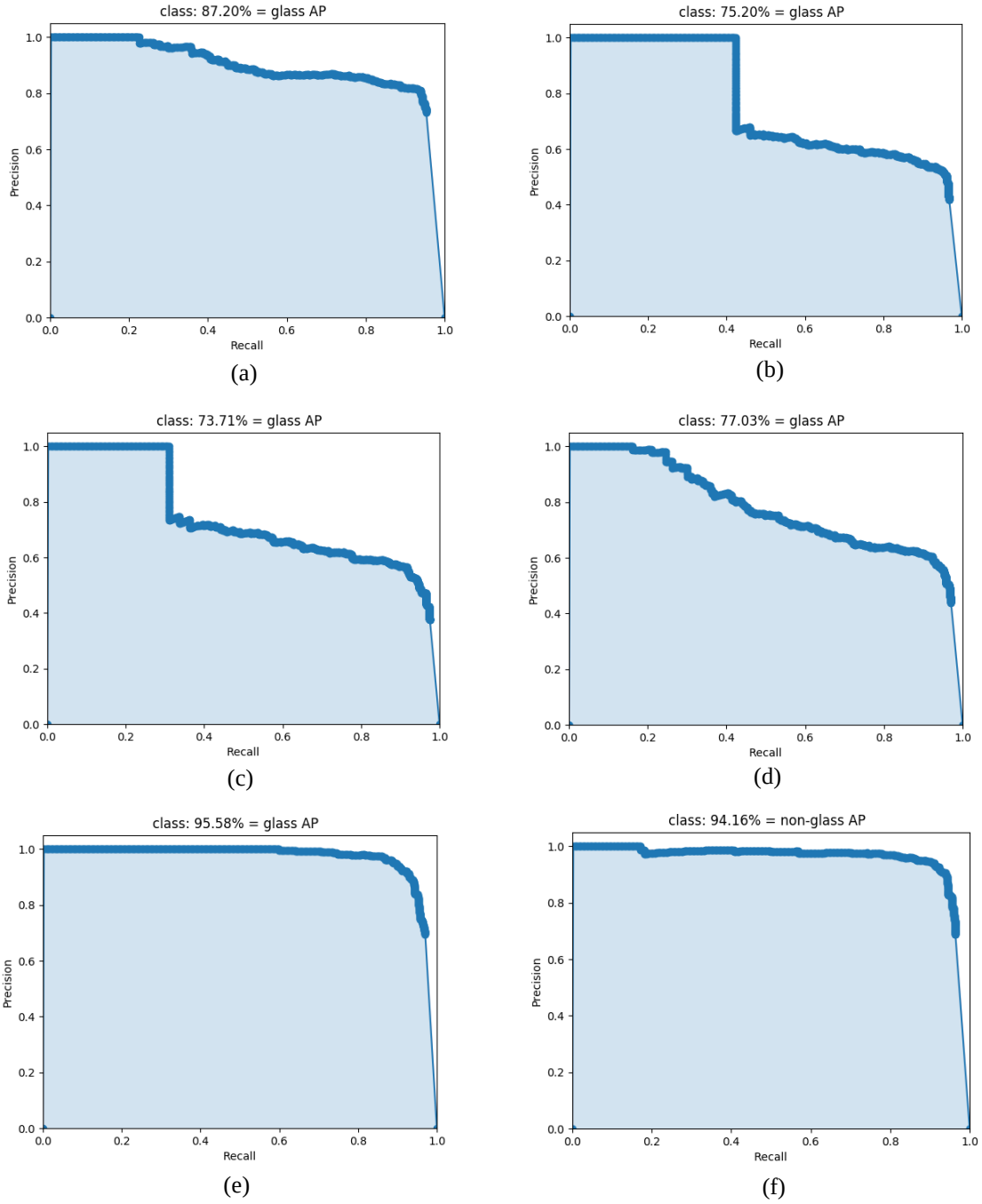


Figure 18. Precision-recall curves and AP results of the network trained with (a) glass and glass-feature (proposed method), (b) only glass, (c) glass and augmented data, (d) glass and negeative training data, and, (e)(f) glass and non-glass

According to precision-recall curves in Figure 18, all training processes achieves over 73.00 % of average percision in the detection of transparent objects. When we compare the AP our proposed network to APs of (b), (c) and (d), our proposed method gets a higher AP result of over 10.00 %. Although our method has a lower AP compared to AP of (e), AP of our proposed network is reasonable because our method takes a less assumption of training data compared to the network trained with both glass and non-glass data. The AP of non-glass of the network trained with glass and non-glass is shown in (f) because mAP calculation in the next step requires the AP of each class of the trained network.

5.4.4 mean Average Precision (mAP) Comparison

Table 5. mean Average Precision (mAP) of different training processes.

Network		mAP
Network trained with only glass		75.27 %
Network trained with glass and augmented training data		73.71 %
Network trained with glass and negative training samples		77.03 %
Network trained with glass and non-glass		**94.87 %**
Network trained with glass and glass-feature	Glass-feature th 0.0	77.67 %
	Glass-feature th 0.1	84.96 %
	Glass-feature th 0.2	86.23 %
	Glass-feature th 0.3	**87.60 %**
	Glass-feature th 0.4	87.13 %
	Glass-feature th 0.5	85.65 %
	Glass-feature th 0.6	84.19 %
	Glass-feature th 0.7	82.65 %

Finally, the mAP of the networks are calculated and shown in Table 5. According to the mAP results, the network trained with glass and non-glass gets the highest mAP result. Among other networks which do not use the non-transparent object of the same shape training data, the network trained with glass and glass-feature outperforms the other 3 networks: the network trained with only glass, the network trained with augmented glass data and the network trained with glass and negative training samples from 9 classes. The network trained with glass and glass-feature achieves the second highest mAP result when the glass-feature threshold is set to 0.3. Comparing to the network trained with glass and non-glass, the mAP difference is not too much high. From the training data point of view, the network trained with glass and non-glass consumes much more traininig data than the network trained with glass and glass-feature. Therefore, the mAP result of the network trained with glass and glass-feature can be said that it gives a considerable performance just with a small amount of training data.

To conclude the comparison between the networks, the network trained with glass and non-glass has the highest mAP results but it totally depends on the availability of negative

non-transparent data in its training process. When we compare this network with our proposed network trained with glass and glass-feature, our method achieves a near mAP performance (just 7.27 % difference) with a much lower cost in selecting training data. Therefore, from training data assumption and precsion-recall point of view, our proposed method has a higher performance than the network trained with glass and non-glass. Our proposed method gets a higher mAP and precision results than the other three methods and achieves the highest F-measure among all comparison methods.

CHAPTER 6. CONCLUSION

In this research, we propose transparent object feature region to eliminate the false detections in the detection of transparent object. Instead of using traditional computer vision algorithms for object detection, one of the deep learning models, called Single Shot MultiBox Detector (SSD) [7], is applied to create a convolutional neural network for the detection of transparent objects. Although SSD gives a high performance in the detection of transparent objects, it shows many false detections when testing the network on non-transparent objects with the same shape of the transparent objects. As a result, the performance of the network is degraded by many false detections. Therefore, object feature region over the transparent objects is introduced to deal with the false detections problem of the network.

The transparent object feature region that we proposed in this research is based on the transparent property of the glass objects and we call these regions as 'glass-feature regions'. These glass-feature regions are appeared on the glass objects because everything inside or outside can be seen through the glass body. We define at least one glass-feature regions on each transparent objects and include them in training the network. Then, the false predictions are removed by defining the regions without any glass-feature as the non-transparent objects. With this approach, a large number of falsely detected regions are eliminated during testing and the network achieves a higher accuracy performance in the detection of transparent objects.

Currently, the proposed transparent object feature region varies according to each kind of transparent objects and we need to define different feature regions for every kinds of transparent objects. In the future work, it will be better if we can find another visual property that is common to all kinds of transparent objects and can be included in the training processes.

REFERENCES

[1] McHenry, K., Ponce, J., Forsyth, D.: Finding glass. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 973-979 (2005).

[2] Deep learning https://en.wikipwdia.org/wiki/Deep_learning.

[3] Everingham, M., Gool, L. V., Williams, C. K., Winn, J., and Zisserman, A.: The Pascal Visual Object Classes (VOC) Challenge. International Journal of Computer Vision, pp. 303-338 (2010).

[4] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollr, P., and Zitnick, C. L.: Microsoft COCO: Common objects in context. In: European Conference on Computer Vision, pp. 740-755 (2014).

[5] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., Fei-Fei, L., ImageNet: a large-scale hierarchical image database. In: IEEE International Conference on Computer Vision and Pattern Recognition, pp. 248-255 (2009).

[6] Adil Moujahid.: A Practical Introduction to Deep Learning with Caffe and Python. adilmoujahid.com/posts/2016/06/introduction-deep-learning-python-caffe/ (2016).

[7] Liu, W. et al.: SSD: Single shot multibox detector. In: Proc. European Conference on Computer Vision 2016, LNCS, vol. 9905, pp. 21-37. Springer (2016).

[8] Adit Deshpande.: A Beginner's Guide To Understanding Convolutional Neural Networks. University of California, Los Angeles (UCLA) (2016).

[9] Transparency and translucency https://en.wikipedia.org/wiki/Transparnecy_and_translucency.

[10] Krizhevsky, A., Sutskever, I., and Hinton, G. E.: ImageNet classification with deep convolutional neural networks. In Neural Information Processing Systems, pp. 1106-1114 (2012).

[11] Zeiler, M. D., and Fergus, R.: Visualizing and understanding convolutional networks. In: Proc. European Conference on Computer Vision, pp. 818-833 (2014).

[12] Simonyan, K., and Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014).

[13] LeCun, Yann, et al.: Gradient-based learning applied to document recognition. Proceedings of the IEEE vol. 86, issue 11, pp. 2278-2324 (1998).

[14] He, K., Zhang, X., Ren, S., Sun, J. Deep residual learning for image recognition. In Proceedings of Computer Vision and Pattern Recognition, pages 770-778 (2016).

[15] Girshick, R., Donahue, J., Darrell, T., and Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 580-587 (2014).

[16] https://blog.heuritech.com/2016/02/29/a-brief-report-of-the-heuritech-deep-learning-meetup-5/

[17] Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. arXiv preprint arXiv: 1506.01497 (2015).

[18] Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. arXiv preprint arXiv: 1506.02640 (2015)

[19] Osadchy, M., Jacobs, D., Ramamoorthi, R.: Using specularities for recognition. In: IEEE International Conference on Computer Vision, pp. 1512-1519 (2003).

[20] Fritz, M., Bardski, G., Karayev, S., Darrell, T., and Black, M.: An additive latent feature model for transparent object recognition. In: Neural Information Processing Systems, pp. 558-566 (2009).

[21] Phillips, C. J., Derpanis, K. G., Daniilidis, K.: A novel stereoscopic cue for figure-ground segregation of semi-transparent objects. In: IEEE International Conference on Computer Vision, pp. 1100-1107 (2011).

[22] Lysenkov, I., Eruhimov, V., Bardski, G.: Recognition and pose estimation of rigid transparent objects with a kinect sensor. In: Robotics, Science and Systems, p. 273 (2013).

[23] Lysenkov, I., Rabaud, V.: Pose estimation of rigid transparent objects in transparent clutter. In: IEEE International Conference on Robotics and Automation, pp. 162-169 (2013).

[24] Xu, Y., Nagahara, H., Shimada, A., Taniguchi, R.: TransCut: Transparent object segmentation from a light-field image. In: IEEE International Conference on Computer Vision, pp. 3442-3450 (2015).

[25] Lai, P. J., Fuh, C. S.: Transparent object detection using regions with convolutional neural network. In: IPPR Conference on Computer Vision, Graphics, and Image Processing, pp. 1-8 (2015).

[26] Uijlings, J. R., van de Sande, K. E., Gevers, T., and Smeulders, A. W.: Selective search for object recognition. In: International journal of computer vision, vol. 104, pp. 154-171 (2013).

[27] Khaing, M. P., Masayuki, M. Transparent object detection using convolutional neural network. In: International Conference on Big Data and Deep Learning, pp. 86-93 (2018).

[28] https://github.com/weiliu89/caffe/tree/ssd

[29] cs.unc.edu/~wliu/projects/ParseNet/VGG_ILSVRC_16_fc_reduced.caffemodel.

[30] Jaccard index https://en.wikipedia.org/wiki/Jaccard_index.