Small-Object Sensitive Segmentation and Quantitative Analysis of Hard Drive Disk

Hao-yu Liao
Environmental Engineering Sciences
University of Florida
haoyuliao@ufl.edu

Xinyao Zhang
Environmental Engineering Sciences
University of Florida
xinyaozhang@ufl.edu

Shuaizhou Hu
Environmental Engineering Sciences
University of Florida
s.hu@ufl.edu

Abstract

We present sensitive semantic segmentation algorithms to detect small components of hard drive disk (HDD). Recent advancements in deep learning have shown an exciting promise in daily-life scene segmentation. In contrast with the larger objects' segmentation approaches, we propose visual attention to segment small-sized interesting objects. Firstly, we introduce a data preprocessing module to mark each component of an HDD. One experimental HDD contains five different shapes of parts, and these multiple classes in the original RGB image dataset are preprocessed with the customized markers labeled in different colors. Secondly, we design deep learning models to classify each component. The GoogLeNet and ResNet-50 models use preprocessed mask images to predict multiple labels. Lastly, we propose multi-region deep convolutional neural networks (CNNs) to predict semantic segmentation results. The resulting CNN-based representation aims to capture a diverse set of areas with inconspicuous appearance and exhibit corresponding localization. We also exploit quantitative analysis of segmentation results by integrating segmented images on two classification models. The final results show the semantic segmentation information of the HDD dataset and the accuracy of multi-region segmentation.

1 Introduction

Large-scale images have become widely available due to the convenience of social media websites and the wide use of digital devices. These images are often attached with rich multimedia information such as tags, comments, etc., which sparks the growing interest in understanding these images by assigning semantic annotations. Semantic segmentation aims to set a categorical label to each pixel in an image, and it plays a vital role in image understanding. The recent success of CNNs has made remarkable progress in pixel-level semantic segmentation tasks.

However, there are two challenges in CNN-based semantic segmentation networks: (1) consecutive pooling or striding causes the reduction of the feature resolution; (2) the networks are not aware of small objects. The segmentation of small objects is usually inaccurate, as small objects typically contribute less to the segmentation loss. For example, as shown in Figure 1, HDD is a small object in real life, and further, the size of its components is smaller, which could be overlooked in the segmentation. However, accurately segmenting small objects is of great importance in many applications, such as autonomous remanufacturing. Proper segmenting and sorting of the waste stream is a primary step in efficiently recovering and handling used products.

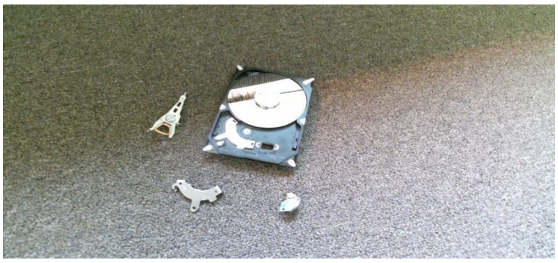


Figure 1: *An example of an HDD image.*

We address the problem of improving the segmentation accuracy of small objects by implementing them in CNNs. The approach does require data augmentation or an increase of the feature dimension. We have reduced the workload about increasing the scale of input images, aiming to enhance the resolution of small objects or produce high-resolution feature maps. Simply increasing the scale of input images often results in heavy time consumption for both training and testing. We implement the strategy to train the network to generate multi-region representation, which enhances high-level small-scale features with multiple low-level feature layers. Another strategy is postprocessing towards improving the accuracy of small object segmentation. Postprocessing is not integrated into the segmentation network; the network cannot update its weights according to the post-processed results in the training phase.

Besides semantic segmentation tasks, image classification as a fundamental multimedia problem has been comprehensively studied for decades, especially for the single-label classification, various progress has been made on it. However, the real-world social image usually contains abundant semantic information, such as objects, attributes, actions, and scenes, etc. With the availability of large-scale images and the enrichment of the meta-data annotations, multi-label image classification has drawn lots of attention. Inspired by the advanced performance of CNNs, various efforts have been made to apply the neural network to multi-label classification problems.

We use GoogLeNet and ResNet-50 models to classify multiple components of HDD. By assigning multiple labels to an image, we can transfer the visual information to language, which is more convenient to understand. Most importantly, we can transfer semantic segmentation results to classifying models, which can be better handle the key issue of segmentation accuracy. We bridge the gap existing in how to quantify the segmentation accuracy. As we can see, there exists not only semantic visual relevance but also image-label accuracy dependencies within semantic regions.

2 Previous Work

GoogLeNet: GoogLeNet is another architecture used in this paper, as shown in Figure 2. It was first proposed by Szegedy et al. in 2014 [1]. It is the first version of Inception, namely Inception-v1. Many researchers used GoogLeNet to classify images in different applications. Singla et al. (2016) applied GoogLeNet to identify food images using transfer learning [2]. Lee et al. (2018) used it to improve the performance of recognition on Korean characters [3]. Jahandad et al. (2019) created an offline signature verification system by using GoogLeNet [4]. The GoogLeNet was trained by over a million images with 1000 different objects and has 22 layers (27 layers considering pooling layers. GoogleNet uses the global average pooling at the last inception module.

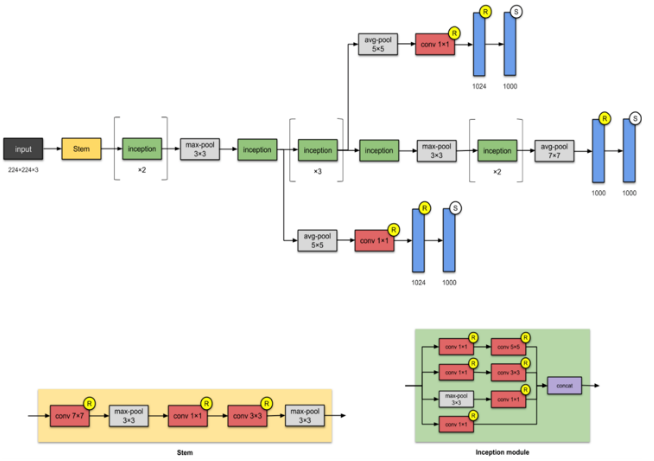


Figure 2: *Architecture of GoogLeNet (Inception-VI) obtained from ref* [5] *based on* [1].

ResNet-50: Stacking layers and making deeper networks will cause the accuracy to get saturated and degrade rapidly [6]. To address this issue, Microsoft Research launched Residual Neural Network (ResNet) in 2015, as a novel architecture with "shortcut connections" (or skip connections, residuals) [6] that skips one or more layers and has the heavy batch normalization function [7] in building models deeper.ResNet is a deeper trained Neural Network while maintaining lower complexity compared to VGGnet [8]. The original models (ResNet-50, ResNet-101, and ResNet-152) were used in ILSVRC and COCO 2015 competitions, which won the 1st places in: ImageNet classification, ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation [6].

Semantic Segmentation: Semantic segmentation that allocates a semantic label to each pixel in an image is one of the most challenging tasks in computer vision. However, traditional image segmentation methods [9], [10], [11] are hard to address the task since they segment objects without "semantic" information. Convolutional Neural Networks (CNNs) [12] provide a breakthrough for semantic segmentation tasks. Thanks to advances of CNNs, recent works achieve highly accurate results even in complex images. Fully Convolutional Networks (FCNs) based on the CNN architecture are widely used thanks to their outstanding performance on semantic segmentation. FCN is aimed to address the problems by using an encoder-decoder structure. The decoder part recovers the object details and spatial information.

3 Overview

In this project, we focus on segmenting multiple small regions from a database of HDD images. We describe the datasets we operate, different classification models, and the final semantic segmentation model. To sum up, the main achievements of our project are: 1) We develop a preprocess network that yields a customized region representation capable of labeling in different colors. 2) We furthermore apply regional mask images on two deep learning architectures for the multi-label image classification, which effectively captures the latent semantic dependencies at the regional level. 3) We show how the segmentation model works and how to significantly quantify the segmentation capability, adopt a CNN model, and quantify segmenting results through classification architectures.

3.1 Dataset

The original HDD dataset was captured by Meng-Lun Lee at the University of Buffalo, which contains 75 images. Some image examples are shown in Figure 3. As our goal is to detect different small regions for an HDD, we separate an HDD into five parts, including hard_disk, disk, y_part, reader, and chip. Some parts, such as the hard_disk base and disk, have a large size, but other remaining parts are relatively small, as shown in Figure 4. Thus, the size of dataset is 375 images (5 times of 75) with each component has the same dataset size as original set. 90% of the dataset is used for training, and 10% for testing. However, the previous dataset does not include multiple components mask images with the ground truth. We create a preprocess network to obtain the segmentation ground truth labeling, as shown in Figure 5.

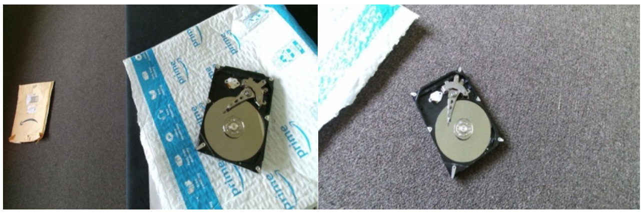


Figure 3: *Different angles, sizes, and views of images are considered when training deep learning architectures.*

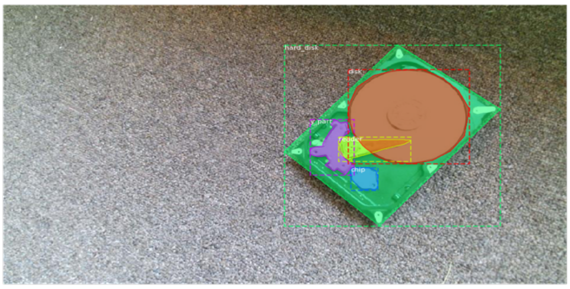


Figure 4: *An HDD is separated into five parts, which are hard_disk, disk, y_part, reader, and chip.*

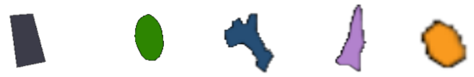


Figure 5: *Segmented images with ground truth from left to right are followed by hard_disk, disk, y_part, reader, and chip.*

3.2 Experimental Design

In order to expand the size of dataset, data augmentation [13] was adopted by resizing, flipping horizontally and normalizing the original data. Two deep architectures, GoogLeNet and ResNet50 were applied. Pretrained models of two architectures are publicly available, coming along with utilizing transfer learning [14] to modify parameters to operate the multi-label image classification. Integrated by segmented images, two deep architectures with trained parameters constituted as semantic segmentation model, which completed segmented prediction task with accuracy of multi-region segmentation information as the output.

4 Methodology

4.1 Technical Description

We use Anaconda 64-bit to build the environment for applying CUDA to run GPU and used the Jupyter notebook from Anaconda to create Pytorch programs. To start this project, the preliminary is dataset processes. First, to process the HDD dataset into mask images, in which we define five small components and use them to modify a larger image. The second is converting input images and mask images into NumPy arrays. Two multi-region CNN architectures are adopted to classify GoogLeNet and ResNet50, with 100 epochs train parameters from both architectures. For the classification procedure, each HDD component will be classified individually, and results will be plotted in confusion matrixes to compare. For the image segmentation with the semantic segmentation model, the trained parameters of GoogLeNet and ResNet50 will be used in the semantic segmentation model with trained parameters of 10000 epochs to output images with each component mask, and this output mask from the segmentation part will become the input to both classification models. With integrating segmented images, two deep architectures will predict the probability of exploring multi-region segmentation accuracy.

4.2 Results & Analysis

This project has two main tasks: classification and segmented prediction, containing three models, GoogLeNet, ResNet50, and the semantic segmentation model. The performance of the two classification models is analyzed as shown in Table 1; GoogLeNet and ResNet50 are comparable. Figures 6 to 9 show the normalized confusion matrix of GoogLenet and ResNet50 model, the training accuracy of two architecture are equally matched. Both models have above 90% accuracy in chip training, above 81% accuracy in disk training, and 75% in reader training; the trade-off is hard-disk and y-part training. Each architecture has better accuracy than the other with one of the above components. And for the testing accuracy, these two architectures are the same with stunning 100%. This phenomenon is because of the limitation on the number of the dataset (75 images). However, five components can help expand the dataset five times; for each component, the number of the dataset is equal to the input.

***Table 1**:* The training and testing results of each component of two architectures.

Model	Parts	Training Accuracy	Testing Accuracy
GoogLeNet	chip	0.99	1
	disk	0.81	1
	hard-disk	0.93	1
	reader	0.76	1
	y-part	0.67	1
ResNet50	chip	0.91	1
	disk	0.85	1
	hard-disk	0.76	1
	reader	0.75	1
	y-part	0.76	1

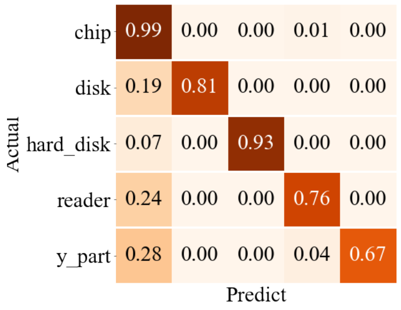


Figure 6: *The training results of classification with normalized confusion matrix of GoogLeNet.*

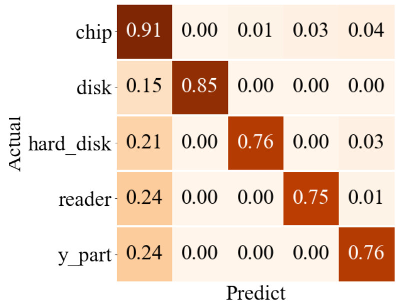
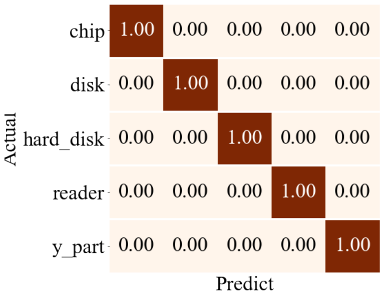
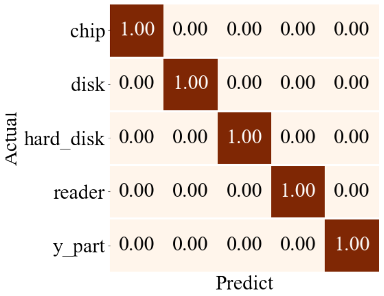


Figure 7: *The training results of classification with normalized confusion matrix of ResNet50.*



Actual \ Predict	chip	disk	hard_disk	reader	y_part
chip	1.00	0.00	0.00	0.00	0.00
disk	0.00	1.00	0.00	0.00	0.00
hard_disk	0.00	0.00	1.00	0.00	0.00
reader	0.00	0.00	0.00	1.00	0.00
y_part	0.00	0.00	0.00	0.00	1.00

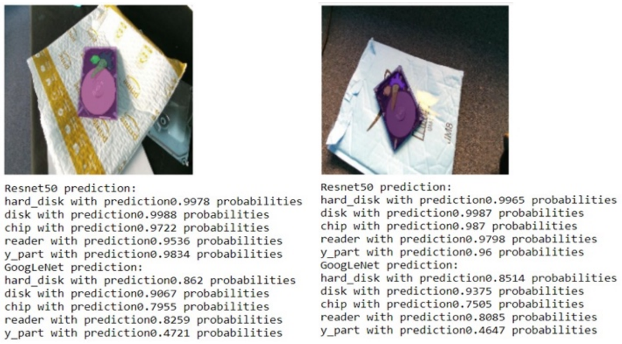
***Figure 8:** The testing results of classification with normalized confusion matrix of GoogLeNet.*



Actual \ Predict	chip	disk	hard_disk	reader	y_part
chip	1.00	0.00	0.00	0.00	0.00
disk	0.00	1.00	0.00	0.00	0.00
hard_disk	0.00	0.00	1.00	0.00	0.00
reader	0.00	0.00	0.00	1.00	0.00
y_part	0.00	0.00	0.00	0.00	1.00

***Figure 9:** The testing results of classification with normalized confusion matrix of ResNet50.*

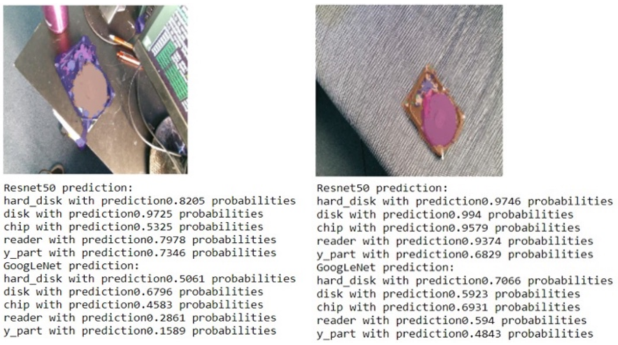
The results of segmented prediction are two random image results picked from the training program and two picked from the testing program; as shown in Figure 10 and Figure 11, ResNet50 is significantly outperforming to GoogLeNet. Despite ResNet50 with higher testing accuracy for every component than GoogLeNet. For segmented image prediction, ResNet50 can predict disk parts with the highest accuracy among all parts above 97%. In comparison, GoogLeNet with great lower accuracy of 60% approximately, for the hard-disk prediction, ResNet50 also has 25% accuracy higher than GoogLeNet. In reader and y-part prediction, ResNet50 shows significant advancement to GoogLeNet again. But in chip prediction, the two results of ResNet50 are far different, which are 53.25% and 95.79%, but when reviewing the test accuracy of 97% higher, we can obtain this phenomenon is not universal. The reason is that ResNet50 is more complicated and advanced than GoogLeNet. ResNet50 has 26 million parameters and 50 layers, while GoogLeNet only has 5 million parameters with 22-layer stacked. And the research paper of ResNet50 is one year later published than GoogLeNet in 2015; the better performance is reasonable. In practical, ResNet50 is a better choice for this kind of task.



```
Resnet50 prediction:
hard_disk with prediction0.9978 probabilities
disk with prediction0.9988 probabilities
chip with prediction0.9722 probabilities
reader with prediction0.9536 probabilities
y_part with prediction0.9834 probabilities
GoogLeNet prediction:
hard_disk with prediction0.862 probabilities
disk with prediction0.9067 probabilities
chip with prediction0.7955 probabilities
reader with prediction0.8259 probabilities
y_part with prediction0.4721 probabilities
```

```
Resnet50 prediction:
hard_disk with prediction0.9965 probabilities
disk with prediction0.9987 probabilities
chip with prediction0.987 probabilities
reader with prediction0.9798 probabilities
y_part with prediction0.96 probabilities
GoogLeNet prediction:
hard_disk with prediction0.8514 probabilities
disk with prediction0.9375 probabilities
chip with prediction0.7505 probabilities
reader with prediction0.8085 probabilities
y_part with prediction0.4647 probabilities
```

***Figure 10:** Two randomly picked training results of segmentation*



```
Resnet50 prediction:
hard_disk with prediction0.8205 probabilities
disk with prediction0.9725 probabilities
chip with prediction0.5325 probabilities
reader with prediction0.7978 probabilities
y_part with prediction0.7346 probabilities
GoogLeNet prediction:
hard_disk with prediction0.5061 probabilities
disk with prediction0.6796 probabilities
chip with prediction0.4583 probabilities
reader with prediction0.2861 probabilities
y_part with prediction0.1589 probabilities
```

```
Resnet50 prediction:
hard_disk with prediction0.9746 probabilities
disk with prediction0.994 probabilities
chip with prediction0.9579 probabilities
reader with prediction0.9374 probabilities
y_part with prediction0.6829 probabilities
GoogLeNet prediction:
hard_disk with prediction0.7066 probabilities
disk with prediction0.5923 probabilities
chip with prediction0.6931 probabilities
reader with prediction0.594 probabilities
y_part with prediction0.4843 probabilities
```

***Figure 11:** Two randomly picked testing results of segmentation.*

4.3 Applications

With correct recognition and location of the different components in HDD, this project is the prerequisite for the application of a robot ta o disassemble and assemble it. This method can also be applied not limited to HDD in the smartphone, laptop, monitor, etc. And this project can be extended in numerous aspects. First, we can detect and locate broken components to help the fixing procedure more efficient. Second, the real-time monitor can save time and avoid worse outcomes for industry, hospital, or lab. Third, valuable, recyclable, and hazardous components can be collected for money-saving and environmental protection.

5 Conclusion

5.1 Discussion

In this project, two deep learning and one segmentation technique, including GoogLeNet, ResNet50, and the semantic segmentation model, are applied to classify and detect five small components of the hard drive disk (HDD). The results showed that, in general, ResNet50 is a better selection than GoogLeNet in this kind of task because ResNet50 is more complicated and advanced than GoogLeNet. But the predicted accuracy of some components is not ideal, which is due to the limited dataset. On the other hand, the high accuracy showed this algorithm is feasible in real life, and it can be extended to different fields.

5.2 Future Work

The limitation of the project is the algorithm is only applied to detect components of the HDD without detecting existing screws connected to these components, so the proposal of achieving disassemble and assemble robotically is incomplete; we must dismantle the screws first and let the robot finish the rest of the task. But screws are much smaller than the smallest components of the HDD; a significant decrease in predicted accuracy is foreseeable if nothing changes in the algorithm. For future work, we will try to find a feasible solution and build an ideal model to solve this challenge, and let the whole process be completely automatic.

References

[1] C. Szegedy *et al.*, “Going deeper with convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[2] A. Singla, L. Yuan, and T. Ebrahimi, “Food/non-food image classification and food categorization using pre-trained googlenet model,” in *Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management*, 2016, pp. 3–11.

[3] S.-G. Lee, Y. Sung, Y.-G. Kim, and E.-Y. Cha, “Variations of AlexNet and GoogLeNet to improve Korean character recognition performance,” *J. Inf. Process. Syst.*, vol. 14, no. 1, pp. 205–217, 2018.

[4] S. M. Sam, K. Kamardin, N. N. A. Sjarif, and N. Mohamed, “Offline signature verification using deep learning convolutional neural network (CNN) architectures GoogLeNet inception-v1 and inception-v3,” *Procedia Comput. Sci.*, vol. 161, pp. 475–483, 2019.

[5] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[6] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[7] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *International conference on machine learning*, 2015, pp. 448–456.

[8] A. Dhillon and G. K. Verma, “Convolutional neural network: a review of models, methodologies and applications to object detection,” *Prog. Artif. Intell.*, vol. 9, no. 2, pp. 85–112, 2020, doi: 10.1007/s13748-019-00203-0.

[9] O. J. Tobias and R. Seara, “Image segmentation by histogram thresholding using fuzzy sets,” *IEEE Trans. Image Process.*, vol. 11, no. 12, pp. 1457–1465, 2002, doi: 10.1109/TIP.2002.806231.

[10] R. Muthukrishnan and M. Radha, “Edge Detection Techniques For Image Segmentation,” *Int. J. Comput. Sci. Inf. Technol.*, vol. 3, no. 6, pp. 259–267, 2011, doi: 10.5121/ijcsit.2011.3620.

[11] F. C. Monteiro and A. Campilho, “Watershed framework to region-based image segmentation,” *Proc. - Int. Conf. Pattern Recognit.*, pp. 0–3, 2008, doi: 10.1109/icpr.2008.4761587.

[12] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998, doi: 10.1109/5.726791.

[13] D. A. Van Dyk and X.-L. Meng, “The art of data augmentation,” *J. Comput. Graph. Stat.*, vol. 10, no. 1, pp. 1–50, 2001.

[14] L. Torrey and J. Shavlik, “Transfer learning,” in *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, IGI global, 2010, pp. 242–264.