



Vilnius University
Institute of Data Science and
Digital Technologies
L I T H U A N I A



INFORMATICS (N009)

LOW ENTROPY MODEL FOR RECOGNIZING OBJECT VISIBILITY IN IMAGES WITH OCCLUDED OBJECTS OF INTEREST

Bernardas ČIAPAS

October 2020

Technical Report DMSTI-DS-N009-20-12

VU Institute of Data Science and Digital Technologies, Akademijos str. 4, Vilnius
LT-08412, Lithuania

www.mii.lt

Abstract

Real world images often contain partially occluded objects of interest. For example, images from retail store self checkout area often contain products that are covered by a customer's body parts, are placed inside semi-transparent plastic bags, include intensive glare, or some combination of these. The more occluded an object of interest is - the more challenging the recognition task. In order to categorize objects of interest in images with partially occluded objects, the first step is to decide if an image contains enough information about the object of interest in order to be categorized.

The most famous computer vision datasets - such as Imagenet, CIFAR, MNIST - are made of images that contain clearly distinctive objects and are labelled with binary information about level of occlusion: either an object of interest exists in the image and is clearly visible, or not. Such binary labels are not fit for solving the recognition task of object occlusion level.

In this study authors aim to categorize images into [not] containing enough information about objects of interest in order to be categorized. Authors analyze a dataset collected in a real retail store self checkout area where objects of interest are various products. The proposed method uses 6 categories of occlusion variously grouped.

Authors received <0.4 entropy in our best model separating images into visible/invisible categories. The proposed method is practical in applications aiming to separate out images with [not] enough information about objects of interest.

Keywords: self checkout, computer vision, image classification

Contents

1	Introduction	4
1.1	Self checkout image specifics.....	6
2	Literature review	7
3	Methods	9
3.1	Dataset	9
3.2	Image preprocessing	10
3.3	Architecture.....	10
3.4	Model training	11
3.5	Experiments.....	11
4	Results	12
5	Conclusion	15
	References	15

1 Introduction

Self checkout context and issues

Self checkout machines were introduced in retail stores as a means to reduce need of cashiers and to shorten customer checkout time. However, self checkouts raised new problems to retailers: theft and long selection time of barcode-less products. Malignant customers use self checkouts in a variety of ways: they replace barcodes of expensive products with barcodes of cheaper ones, intentionally pick cheaper products from pick list menu. Benign customers suffer longer checkout times due to having to pick each barcode-less product from picklist menu that contains many similar products, has a hierarchical structure of 3-5 levels. Complex picklist menu often results in unintentional selection of wrong products and need for staff assistance. Retail industry badly needs to solve these problems. Successful solutions would simplify product selection from picklist menu and raise alerts upon scanning/selecting incorrect products. In this research authors analyze a computer vision based approach to recognize products that could address each of the mentioned issues.

Self checkout process

Table 1 shows the flow of products movement during self checkout process. A customer brings a shopping basket (left in the picture) or a trolley full of products to be purchased to the checkout area. Then she takes one product at a time from a basket/trolley registers it in one of two ways: scans (products with barcode stickers - e.g. milk packs) or picks from a menu (barcode-less products - e.g. fruits). Scanner is usually located under the glass (green rectangle in the picture) and/or behind a glass in front of the customer (above the green rectangle in the picture). A picklist menu to select barcode-less products is displayed on a touch screen in front of the customer (above the green rectangle in the picture - not shown). Upon picking a barcode-less product from a menu, it is weighed by scales (green rectangle in the picture). Finally, when product is registered, a customer moves it to the bagging area.

Self checkout data collection challenges

Scanner/scales area (green frame in Table 1) usually contains a single product, while other areas - shopping basket, bagging - usually contain more. Self checkouts register events: scanning of a barcode, weighing a picked from menu barcode-less product. Both at time of scanning and weighing, a product is contained in the green frame. Thus, it was possible to take photos at the moment of scanning/weighing and label them with product ID.

It is a much more complex task to recognize individual products in shopping basket, bagging area - since multiple products are placed there. In terms of computer vision, this would be an object detection task that requires labels with product location bounding boxes. Authors refrain from detection task in basket and bagging areas in this research,



Shopping basket,
multiple products



Scanner/sales area,
single product
(area of research)



Bagging area,
multiple products

Table 1: Checkout flow

although solving it has a variety of applications.

State of the art in computer vision

Convolutional neural networks like (Simonyan and Zisserman, 2015), (He et al., 2016) achieve impressive results on image classification task. Convolutional filters extract relevant object features as shown in (Zeiler and Fergus, 2014), then grouped by dense layers to decide on object class. However, most benchmark datasets (ImageNet, CIFAR[-10]-100], MNIST) only include images where visibility of objects of interest is binary: only images will clearly visible and distinctive objects are included.

The gap and focus of this research

Real life images, which need to be classified, often contain objects that are occluded to some degree. For example, most self checkout images contain products partially covered by a customer's hand or other body part; about 15% of barcode-less products are sold in plastic bags that are semi-transparent; specific locations within the scales area reflect light in a way that reduces recongizability. Since images with more occluded objects are likely to contain less information about the object of interest, simply applying classification techniques on images with occluded objects is likely to result in lower classification metrics. In order to keep satisfactory classification metrics, images with occluded objects must be decided on wheather objects of interest are visible enough for classification.

In this study authors propose a way to classify images with occluded objects of interest into visibility categories. The main criteria evaluated is how well images with similarly occluded objects are classified into a same visibility category. Authors aim to show that object occlusion level can be measured with low entropy.

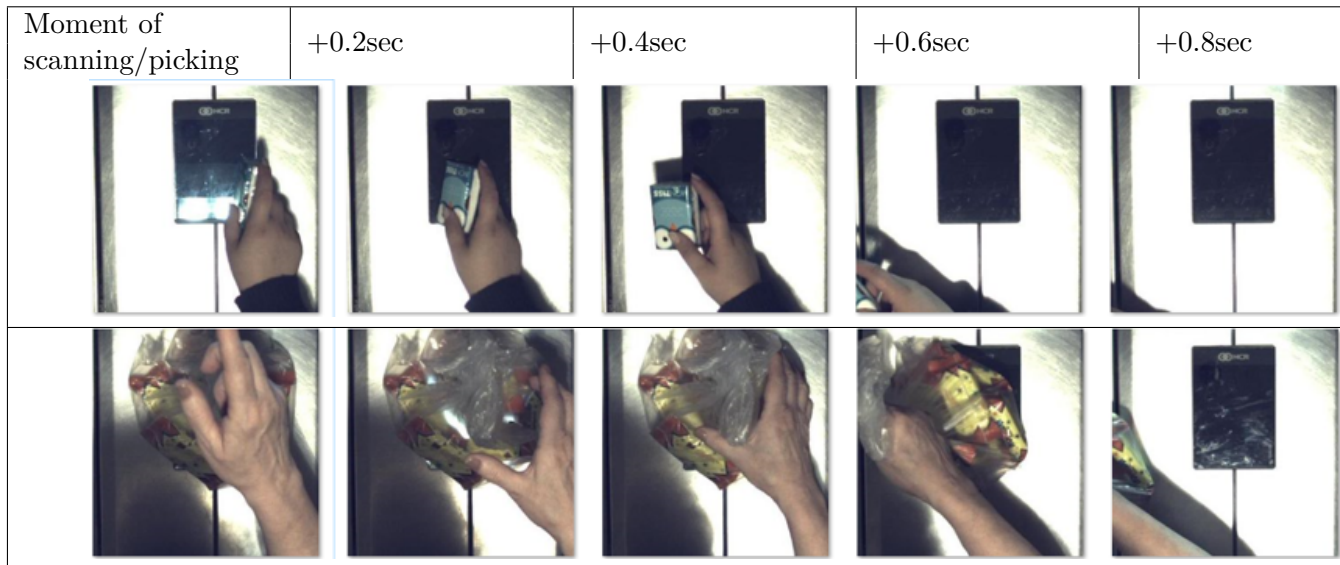


Table 2: Time sequence of scanned (top row) and picked (bottom row) products

1.1 Self checkout image specifics

Self checkout image crops of scanner area taken by an overhead camera have certain specifics:

- Products covered by hand or other body part
- Products packed in plastic bags
- Products of different size
- Static background
- Illumination differences

Products covered by a body part Products with barcodes are usually held in customer's hand at the moment of scanning, then moved to the bagging area. In Table 2 top row, the left frames represent the moment of barcode scanning, and a frame to the right shows +0.2 second increment from it's left neighbour. Depending on the position of a barcode on a product, the moment of scanning may not contain enough product information, like the upper left frame in the sample figure. Later frames (4-5 in shown sample) may have a product completely removed.

Barcode-less products are picked from a touch screen menu after being placed on the scales. At a self checkout's event of picking an item, a customer's hand (and sometimes head) usually covers a part of a product placed on the scales as shown in Table 2 bottom row, left frames. Then a customer usually lowers an arm, thus uncovering a bigger part of a product (frames 2-4), then moves a product to the bagging area.



Figure 1: Bananas: unpacked, in a plastic bag, in a plastic bag with high reflection



Figure 2: Products vary in size

Products in plastic bags Approximately 15% images contain products packed in transparent plastic bags (excl. original producer packing). While sometimes products in bags are recognizable, other times plastic bags reflect light in a way that makes products not visually recognizable, as shown in Figure 1 right frame.

Products of different size Products vary greatly in size as shown in Figure 2. Thus images with smaller products carry less information about the product of interest and more about the background.

Illumination differences Illumination differences occur between different self checkout instances, at different times of day, in different self checkout zones (Table 3).

2 Literature review

Recent advances in covered object recognition use a see-through terahertz beam such as (Wang et al. 2019) and analyze reflection signal amplitude and phase differences in materials. Such terahertz cameras are far from ubiquitous and will hardly ever be, and our method uses a more widespread RGB image features.

Some publicly know datasets such as Imagenet (Deng et al. 2009), Pascal VOC (Ev-

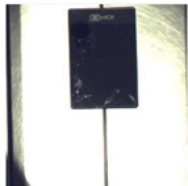




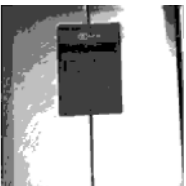
Difference in:					
Self checkout instances		Times of day		Illumination zones (and zone masks)	
					

Table 3: Illumination differences

eringham et al. 2010) use rectangular bounding boxes as ground truth to mark object location and size. Others use even more precise object shape markings: Caltech 101 (Li Fei-Fei et al. 2006), LabelMe (Russell et al. 2008) use closed boundaries and MSRC (Ali and Zafar 2018) uses pixel level segmentation. Each of the above object marking ways - rectangular bounding boxes, closed boundary shapes, and pixel level segments - are costly to label for new datasets. Our method only requires class labels for images, thus making it less costly to label a new domain-specific dataset.

To extract features from images authors train convolutional filters which are class-agnostic, but sensitive to object's existence. Very similar concept - class-agnostic convolutional filters on object-containing widows - was used in (Singh et al. 2018), but authors train on full images rather than object-containing crops (due to this dataset annotation nature). These methods generate region proposals, then extracts features from them: (Russell et al. 2006) extracts visual words from pixel level segments, then compares to those of known object bounding boxes; (Alexe et al. 2012) finds closed boundary shapes. Both of the above methods imply having learnt features from a dataset annotated with object locations, which didn't exist in the dataset used in this research.

Most methods using datasets where object location is defined - (Singh et al. 2018), (Cheng et al. 2019) - use IoU (intersection-over-union) to measure correctness of object localization. Since in this research authors didn't use dataset with annotations of object location, class labels (Is/Isn't an object) were used in measuring correctness.

To evaluate models authors used entropy. Although cross entropy is more widely applied as a loss function in classification tasks since the beginning of artificial neural networks (Long et al., 2016) and (Krizhevsky et al., 2012), but it measures match between 2 populations; as opposed, the goal of this research is only to measure how well a single population is grouped together.

Many researchers use entropy to create unsupervised models: (Yin et al., 2017) attempts to maximize entropy among different image background/foreground pixels; (Kodors, 2019) and (Quinlan, 1986) try to reduce entropy when selecting next features in forming decision tree nodes; (RIKTERS, 2019) use entropy of output by competing translation systems in order evaluate translation quality. Although in this research the authors use supervised models, but the goal - to group the entire population of a single category together - lets use entropy to evaluate the models.

Entropy is widely used in signal pre-processing: (Liutvinavičienė and Kurasova, 2018) measure entropy between audio frames in order to extract time sequences belonging to the same syllable; (Nežerka and Trejbal, 2019) use entropy to segment images. In this research authors settled for manual image labelling, thus making it possible to formulate the task at hand - deciding if an image contains a visible enough object of interest - as a classification task.

As an alternative to entropy authors could have used Gini impurity that gives similar results (Géron, 2019).

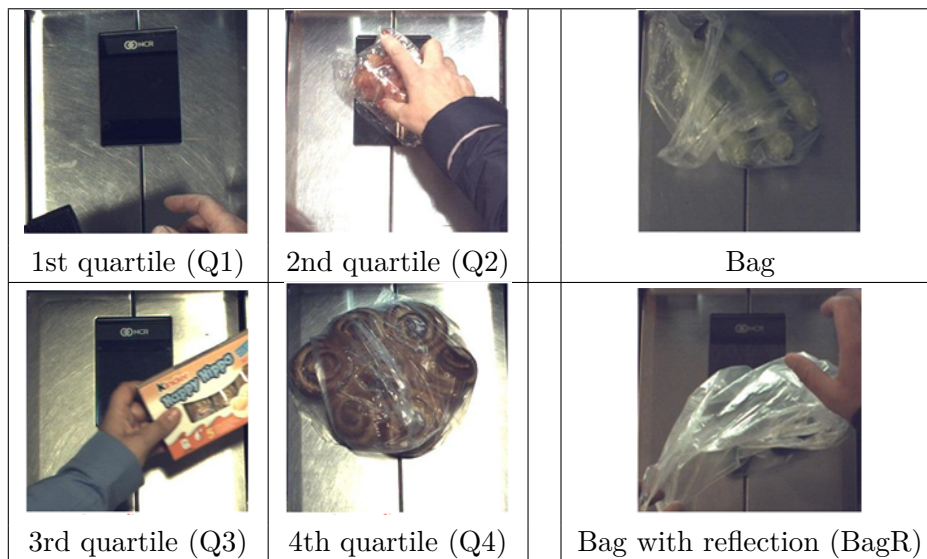


Table 4: Each class samples

3 Methods

3.1 Dataset

Authors have labelled the images into 6 exclusive classes by applying these rules:

- By product visibility quartile (classes Q1–Q4) - for products not in bags
- Products in bags (class Bag) - when product can be recognized by a human
- Products in bags with reflection (class BagR) that makes a product unrecognizable

, samples of each data class are displayed in Figure 4.

The entire labelled dataset consists of 6000 images taken at the time of scanning/picking. Each image represents a different product instance (object of interest). Dataset size was chosen such in order to contain a similar number of samples per class (1000) as Imagenet dataset, where classification task was solved with high accuracy.

Classes are unbalanced as shown in Table 5. Prior to any training, authors balanced the classes by oversampling and then augmenting underrepresented class images. Authors used the following augmentation parameters: rotation (up to 10 degrees), shifting (up to 32 pixels), zoom (up to 10%), and horizontal flip.

Labelling images into a bigger number of product visibility quantiles would have given us more when flexibility splitting data into Visible/Invisible categories. Considering

Q1	Q2	Q3	Q4	Bag	BagR	TOTAL
32%	22%	15%	21%	7.3%	2.6%	100%

Table 5: Image class ratios

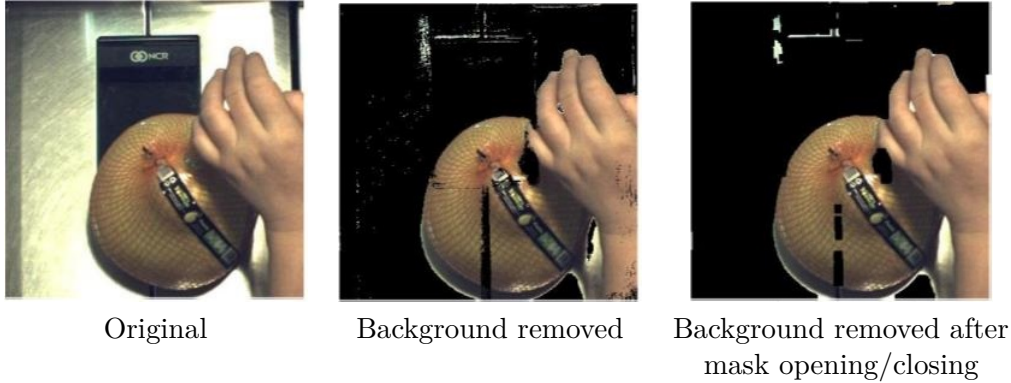


Table 6: Background removed

a human labeller would make more mistakes if more quantiles were used, authors decided to limit the number of visibility quantiles to four.

Images with products packed in plastic bags showed very different features from images with unpacked products in early analysis: plastic bags are easily identifiable, but products inside the bags - not necessarily so. Due to some images with plastic bags having light reflection that makes product unrecognizable, authors decided to split images with plastic bags into classes Bag (recognizable products in plastic bags) and BagR (not recognizable to humans).

3.2 Image preprocessing

Authors used square crops of scanner/scales area as shown in Table 1.

Authors tried removing static background using (Zivkovic and Van Der Heijden, 2006) prior to training. In order to eliminate small foreground patches and fill small foreground mask gaps within products, authors applied morphological opening/closing on background masks. In Figure 6 authors show a sample original image, image with background removed, and image with background removed after having opening/closing operations applied on foreground mask.

To reduce variance in image illumination intensity, authors applied contrast limited adaptive histogram equalization (Zuiderveld, 1994) on HSV "V" channel. In Figure 7 authors show a sample image and contrast limited adaptive histogram equalization applied.



Table 7: Contrast limited adaptive histogram equalization

3.3 Architecture

Authors experimented with classical convolutional neural network architecture (Krizhevsky et al., 2012) by using 1–8 convolutional and 1–3 fully connected layers. Starting with one layer of each type, authors added layers until training accuracy saturated. The best validation accuracy was achieved by using 7 convolutional and 3 dense layer architecture. Authors used convolutional filter size 3x3 - experiments of filter size 5x5, 7x7 showed worse results. Adding batch normalization layers (Ioffe and Szegedy, 2015) after each convolutional and dense generally improved both training and validation accuracy. Adding dropout (Srivastava et al., 2014) after all dense layers helped. Adding L2 regularization helped only after the last dense layer.

3.4 Model training

Authors used dynamic augmentation parameters on training and validation data: rotation (up to 10 degrees), shifting (up to 32 pixels), zoom (up to 10%), and horizontal flip (50% probability). Eliminating either augmentation parameter or reducing parameter range worsened validation accuracy. Increasing augmentation parameter range by 50% made training accuracy below 85%. Removing augmentation just for validation set did not affect accuracy.

Authors stopped training models after validation accuracy did not improve for the last 20 epochs, then reverted parameters to the best epoch's. Additionally training some trained models by halving learning rate generally improved validation accuracy.

3.5 Experiments

Authors mixed data labels in all possible ways into [Visible; Invisible] categories as shown in Figure 3 with the following restrictions:

- Q1 always Invisible;
- Q4 always Visible;

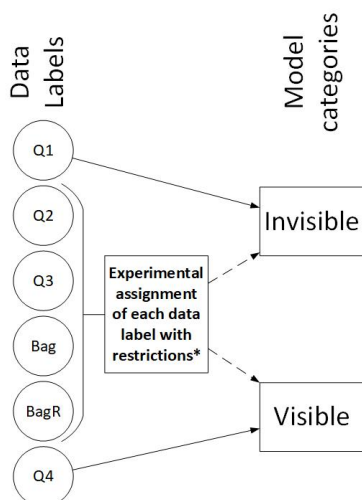


Figure 3: Data labels grouping strategy

- Q3 can only be Invisible if Q2 is not Visible;
- Q2 can only be Visible if Q3 is not Invisible;
- Bag can only be Invisible if BagR is not Visible;
- BagR can only be Visible if Bag is not Invisible.

In all experiments authors used the same number of samples: undersampled data when model category Visible; Invisible contained more the a single label [Q1-Q4,Bag,BagR].

4 Results

Quality of models was evaluated in terms of how well it separates instances of some label Q1-Q4, Bag, BagR into a single category Visible; Invisible: a perfect model would assign all samples of the same label to the same category.

Authors used entropy (1) to measure how well a certain label's test data is assigned into a single model category [Visible; Invisible]. To evaluate models, authors summarized single label entropies into mean weighted entropy of an entire model.

Entropy:

$$H_l(X) = - \sum_{c=1}^C p_c(X_l) \log p_c(X_l), \quad (1)$$

where:

- $l \subseteq [Q1; Q2; Q3; Q4; Bag; BagR]$ - a data label,
- $p_c(X_l)$ - a frequentist probability for a datapoint X_l to be assigned to category c ,

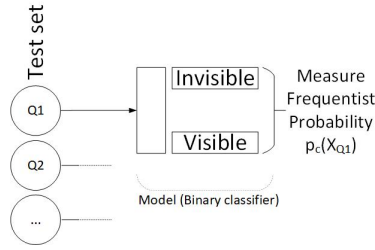


Figure 4: Entropy measurement for each data label

Data Label	Labels in category		Entropy
	Visible	Invisible	
Q1	Q3, Q4	Q1	0.087
Q2	Q4, Bag, BagR	Q1, Q2, Q3	0.213
Q3	Q3, Q4	Q1, Bag, BagR	0.249
Q4	Q2, Q4	Q1	0.149
Bag	Q2, Q3, Q4, Bag, BagR	Q1	0.382
BagR	Q4, BagR	Q1	0.449

Table 8: Least entropy models for each label

- $C = [Visible; Invisible]$ - model categories.

Authors left out of scope of this research to evaluate how accurately objects of interest can be classified in intermediate label images (Q2, Q3, Bag, BagR); however, in order to be useful, model has to split higher vs. lower visibility labels. Authors only report results of "useful" models.

In order to measure entropy of a model for a certain data label, authors fed the entire test set's data of that label into the model, then calculated frequentist probability, as shown in Figure 4.

In table 8 authors present the least entropy models for each data label and their entropies. Higher entropy for less represented data labels (BagR, Bag) and lower entropy got more represented label (Q1) strongly suggests collecting more data for underrepresented labels might help.

Table 9 shows the least mean entropy models for all data labels. The main metric for practical applications - weighted mean entropy <0.4 - suggests that $>92\%$ of a label's data will be split into the same category of [Visible;Invisible].

Mean type	Labels in category		Entropy
	Visible	Invisible	
Mean	Q2, Q4	Q1	0.521
Weighted mean	Q4, Bag, BagR	Q1, Q2, Q3	0.398

Table 9: Least mean entropy models

Least entropy model for Data Label	Model metrics			
	Accuracy	Precision	Recall	F1
Q1	0.713	0.981	0.477	0.642
Q2	0.773	0.815	0.330	0.469
Q3	0.814	0.727	0.960	0.828
Q4	0.858	0.889	0.863	0.876
Bag	0.856	0.908	0.881	0.894
BagR	0.828	0.949	0.625	0.754

Table 10: Least entropy model metrics

Pipeline step	Technique	Impact on accuracy
Balancing classes	Eliminating one augmentation parameter or reducing augmentation range by half	-3.9% – -1.7%
Image pre-processing	Removing static background	-3.6%
	Removing static background after applying morphological operations on mask	-5.4%
	Contrast limited adaptive histogram equalization	+1.4%
Network Architecture	Convolutional filter sizes bigger (5x5, 7x7)	-3.0%
	Batch normalization	-0.6% – +2.2%
	Dropout	+2.2% – +2.5%
Model training	L2 regularization	+2.0%
	Additional training using halved learning rate	-0.1% – +5.7%
	Dynamic augmentation, cutting range by half	-3.4%
	Dynamic augmentation, removing for validation set	0.0%

Table 11: Impact of pipeline techniques tried on validation accuracy

In Table 10 authors present accuracy, precision, recall, f1 scores of the models of least entropy of each data label.

In Table 11 authors summarize the impact of image pre-processing techniques and neural network architecture techniques that were applied in Methods. Impact on validation accuracy shown is relative to the technique omitted from the pipeline.

Authors expected to find positive correlation between number of data labels in both categories and entropy, but observed none: in Figure 5 mean weighted entropy is shown for all experiments per number of data labels in each model category.

In Figure 6 the confusion matrix of the least mean weighted entropy model is displayed.

5 Conclusion

Authors have achieved <0.4 entropy categorizing self checkout images into containing visible/invisible products. Our method is based on a dataset collected in a real retail store’s self checkout area, thus having proportions of various image categories that occur in real world. This study proposes a practical way for applications to decide if a self

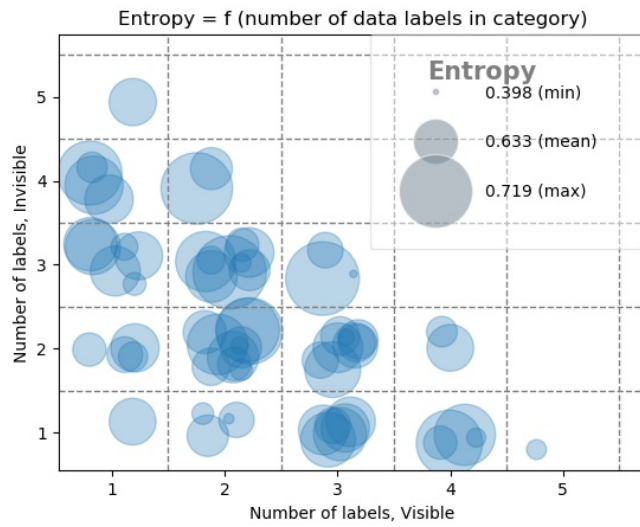


Figure 5: Mean weighted entropy

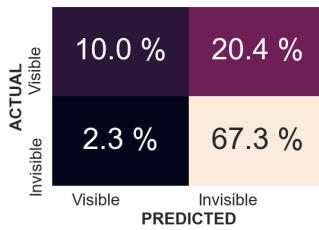


Figure 6: Confusion matrix, least weighted entropy model

checkout image contains a product that can be classified.

References

- B. Alexe, T. Deselaers, and V. Ferrari. Measuring the Objectness of Image Windows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2189–2202, nov 2012. ISSN 1939-3539. doi: 10.1109/TPAMI.2012.28.
- N. Ali and B. Zafar. MSRC-v2 image dataset, aug 2018. URL https://figshare.com/articles/dataset/MSRC-v2_image_dataset/6075788/2.
- M.-M. Cheng, Y. Liu, W.-Y. Lin, Z. Zhang, P. L. Rosin, and P. H. S. Torr. BING: Binarized normed gradients for objectness estimation at 300fps. *Computational Visual Media*, 5(1):3–20, 2019. ISSN 2096-0662. doi: 10.1007/s41095-018-0120-1. URL <https://doi.org/10.1007/s41095-018-0120-1>.
- J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.
- M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) challenge, 2010.
- A. Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O’Reilly Media, 2019.
- K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *2016 {IEEE} Conference on Computer Vision and Pattern Recognition, {CVPR} 2016, Las Vegas, NV, USA, June 27-30, 2016*, volume 2016-Decem, pages 770–778. {IEEE} Computer Society, 2016. ISBN 9781467388504. doi: 10.1109/CVPR.2016.90. URL <https://doi.org/10.1109/CVPR.2016.90>.
- S. Ioffe and C. Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In F. R. Bach and D. M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, {ICML} 2015, Lille, France, 6-11 July 2015*, volume 37 of *{JMLR} Workshop and Conference Proceedings*, pages 448–456. JMLR.org, 2015. ISBN 9781510810587. URL <http://proceedings.mlr.press/v37/ioffe15.html>.
- S. Kodors. Detection of Man-Made Constructions using LiDAR Data and Decision Trees. *Baltic Journal of Modern Computing*, 7(2):255–270, 2019.
- A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. ISBN

9781420010749. doi: 10.1201/9781420010749. URL <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>.

Li Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):594–611, apr 2006. ISSN 1939-3539. doi: 10.1109/TPAMI.2006.79.

J. Liutvinavičienė and O. Kurasova. Multi-level Massive Data Visualization: Methodology and Use Cases. *Baltic Journal of Modern Computing*, 6(4):321–334, 2018. ISSN 2255-8942.

M. Long, H. Zhu, J. Wang, and M. I. Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances in Neural Information Processing Systems*, pages 136–144, 2016.

V. Nežerka and J. Trejbal. Assessment of aggregate-bitumen coverage using entropy-based image segmentation. *Road Materials and Pavement Design*, pages 1–12, 2019. ISSN 1468-0629.

J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986. ISSN 1573-0565. doi: 10.1007/BF00116251. URL <https://doi.org/10.1007/BF00116251>.

M. RIKTERS. Hybrid Machine Translation by Combining Output from Multiple Machine Translation Systems. *Baltic Journal of Modern Computing*, 7(3):301–341, 2019. ISSN 2255-8942.

B. C. Russell, W. T. Freeman, A. A. Efros, J. Sivic, and A. Zisserman. Using Multiple Segmentations to Discover Objects and their Extent in Image Collections. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1605–1614, jun 2006. doi: 10.1109/CVPR.2006.326.

B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. LabelMe: a database and web-based tool for image annotation. *International journal of computer vision*, 77(1-3): 157–173, 2008.

K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, {ICLR} 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, pages 1–14, 2015. URL <http://arxiv.org/abs/1409.1556>.

B. Singh, H. Li, A. Sharma, and L. S. Davis. R-FCN-3000 at 30fps: Decoupling Detection and Classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, jun 2018.

- N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. ISSN 15337928. URL <http://jmlr.org/papers/v15/srivastava14a.html>.
- D. Wang, Y. Zhao, L. Rong, M. Wan, X. Shi, Y. Wang, and J. T. Sheridan. Expanding the field-of-view and profile measurement of covered objects in continuous-wave terahertz reflective digital holography. *Optical Engineering*, 58(2):1–7, 2019. doi: 10.1117/1.OE.58.2.023111. URL <https://doi.org/10.1117/1.OE.58.2.023111>.
- S. Yin, Y. Qian, and M. Gong. Unsupervised hierarchical image segmentation through fuzzy entropy maximization. *Pattern Recognition*, 68:245–259, 2017. ISSN 0031-3203. doi: <https://doi.org/10.1016/j.patcog.2017.03.012>. URL <http://www.sciencedirect.com/science/article/pii/S0031320317301115>.
- M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8689 LNCS(PART 1):818–833, 2014. ISSN 16113349. doi: 10.1007/978-3-319-10590-1_53.
- Z. Zivkovic and F. Van Der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters*, 27(7):773–780, 2006. ISSN 01678655. doi: 10.1016/j.patrec.2005.11.005.
- K. Zuiderveld. Contrast Limited Adaptive Histogram Equalization. In P. S. Heckbert, editor, *Graphics Gems*, pages 474–485. Academic Press, 1994. ISBN 978-0-12-336156-1. doi: 10.1016/b978-0-12-336156-1.50061-6. URL <http://www.sciencedirect.com/science/article/pii/B9780123361561500616>.