DETECTION OF DENSE, OVERLAPPING, GEOMETRIC OBJECTS

Adele Peskin[1], Boris Wilthan[1] and Michael Majurski[2]

[1]NIST, 325 Broadway, Boulder, CO, USA
[2]NIST, 100 Bureau Dr., Gaithersburg, MD, USA

ABSTRACT

Using a unique data collection, we are able to study the detection of dense geometric objects in image data where object density, clarity, and size vary. The data is a large set of black and white images of scatterplots, taken from journals reporting thermophysical property data of metal systems, whose plot points are represented primarily by circles, triangles, and squares. We built a highly accurate single class U-Net convolutional neural network model to identify 97 % of image objects in a defined set of test images, locating the centers of the objects to within a few pixels of the correct locations. We found an optimal way in which to mark our training data masks to achieve this level of accuracy. The optimal markings for object classification, however, required more information in the masks to identify particular types of geometries. We show a range of different patterns used to mark the training data masks, and how they help or hurt our dual goals of location and classification. Altering the annotations in the segmentation masks can increase both the accuracy of object classification and localization on the plots, more than other factors such as adding loss terms to the network calculations. However, localization of the plot points and classification of the geometric objects require different optimal training data.

KEYWORDS

Object detection, Classification, Convolutional Neural Network, Geometric shapes.

1. INTRODUCTION

We were given a collection of thousands of images of plots from old journal articles whose data points were published in pictorial form only and tasked with extracting the location of data points on the plots, so that a facsimile of the raw data can be recovered for additional analysis. The images portray points on the plots with a variety of different geometric markers, circles, triangles, squares, etc., both filled and open versions. It is important to be able to capture the precise locations of these markers, in order to collect this data, not available in any other form. Accurately detecting and localizing these plot markers is different from other object detection problems in several ways. The images are black and white. There is no subtle gray-scale information for a network to utilize. The plot marks are inconsistent in their clarity and exact shape. For example, an open circle could have a well-defined circular shape on its outer boundary, but the inner boundary could be more distorted. The shapes of the objects are all very identifiable and repeatable, so we are only dealing with a very limited geometric scope, and the size of the geometric objects located in these images is well defined within a range of 30-60 pixels. Finally, many of plots contain very dense patches of these geometric objects, so any method we use will need to be robust enough to handle small, overlapping circles, squares, and triangles.

2. RELATED WORK

The detection of dense objects in images has been handled using two basic approaches. The first approach generates large sets of possible object locations, in the form of boundary boxes or sliding windows. The bounding boxes can be defined sparsely, classified as either foreground or background, and then used to classify objects in a two-step approach (as in R-CNN [1] or Fast R-CNN [2]), or defined densely where the network simultaneously assigns a probability of containing an object and the classification of the object in a single step (YOLO [3,4], and SSD [5]). These methods provide a series of solutions which maximize accuracy and/or speed of computation. The output of the network contains information about the location, size, and classification of objects in images.

For dense object detection, different approaches have been used to deal with the problem of overlapping bounding boxes. One approach, discussed in [6], is to add an extra layer to the network (a soft intersection-over-union layer) to provide information on the quality of the detection boxes.

A second path for object detection produces a semantic segmentation of an image, where the output is an image, with each pixel labeled as to its classification. U-Net is an example of this type of convolutional network, which has been used very successfully for a wide range of biomedical image segmentations. U-Net architecture consists of a contracting path, in which spatial information is reduced while feature information is increased, and an expanding path, which combines feature and spatial information, using up-convolutions and concatenations with features of the contracting path [7]. For these networks, the locations of the objects in the images is either output as part of the network in addition to pixel labeling [8] or obtained as a result of processing the output of the network, as in panoptic segmentation, which combines both semantic and instance segmentations [9].

Adding different loss functions into the evaluation of each forward pass of many of these networks has improved the quality of their outcomes. Lin [10] used a focal loss term derived for dense object detection, which adds weight to the relative loss for objects that are harder to classify. In our work, classification problems were not related to specific geometries. Ribera [8] used a loss term that adds weight to the relative loss for pixels based on their distance to objects within the image, reducing contributions from unwanted background pixels.

We approached this problem from the two directions described above, using a YOLOv3 (you only look once) model, which uses bounding boxes [3,4], and a U-Net model, which performs semantic segmentation [7]. Our required accuracy for detecting plot points had to be at least as high as manual detection, within 5 pixels of the actual location on a plot size of several thousand pixels per side.

The YOLO model we used was limited to only a small set of bounding box anchor sizes, given the size range of the objects. However, even when reducing the size of the sub-image for the network search to an 8x8 pixel region, which increased the accuracy of locating the points, it was not enough to meet the 5-pixel requirement. Our best results with the YOLO model located the objects with 10-15 pixels of the manually selected locations, not close enough to satisfy our goal. In addition, we saw a consistently high false positive rate for object detection. The output of a YOLO model contains probabilities that an object lies within each bounding box, and we could not find an adequate threshold to separate the true and false positives. Our U-Net model gave much higher location point accuracy and had a very low false positive detection rate, and we will focus on that model in this paper.

2.1. TRC Data

Our data collection is supplied by the Thermodynamics Research Center (TRC) at the National Institute of Standards and Technology, which curates a database of experimental thermophysical property data for metals systems (https://trc.nist.gov/metals_data). All data are derived from experiments reported in the open literature dating from the early 1900's to very recent publications. While it would be most convenient if the authors had reported tabulated data, often the results are only presented in graphical format and the data must be extracted with digitization tools manually.

This has been done for thousands of plots of varying quality – from figures of very recent papers in a pdf format to hand drawn graphs on grid paper and many levels in between, often also distorted and degraded by low quality scans and photocopies. TRC staff categorized the plots into different groups, single data sets, multiple data sets, with legend, without legend, separated data points, overlapping data points, etc., to provide a comprehensive training data set. The varying clarity of the figures posed many challenges. Some circles appear extremely circular, others do not have enough pixels to fully define their shape. Many of the plots have objects that are separated, and others have masses of overlapping, dense sets of objects. Our approach for object detection successfully covered a wide range of image qualities.

In addition to the actual data points there were also many other symbols an algorithm identifies as data points if not seen in broader context. Text, numbers, and symbols in a legend can cause many false positive results. To eliminate those issues a subset of figures was selected to undergo an extra cleanup step where all misleading symbols were manually removed. Along with each plot, we were given coordinates of the points in picture coordinates.

The raw images are RGB images, mostly black and white, but some have yellowed backgrounds. We applied the following morphological operations to create training and test sets for the networks: (1) convert to grayscale(gs) intensity using the following formula (gs = 0.2989 * red + 0.5870 * green + 0.1140 * blue); (2) binarize grayscale images with a manually defined threshold of 140: all pixels < 140 are converted to 0 and all pixels over 140 are converted to 255; (3) crop images between the minimum and maximum x and y values (as read from the plot points in the images) with a 200 pixel buffer in both directions; (4) break each plot down into overlapping sub-images of 512x512 pixels to feed into the network. Sub-images overlap by 30 %. We chose 512 so that a group of overlapping objects which are on the larger size can appear in the same sub-image.

2.2. U-Net model and segmentation masks

Our initial U-Net network was modeled after the approach of [8], which adds a loss function to the network to remove output other than the objects of interest. An advantage of this model is that it is designed for small objects and for overlapping objects. We found that the accuracy of locating the points was similar with or without the loss function given in [8].

Our U-Net model is shown in Figure 1. Since U-Net requires labeled masks as input, we convert each plot mark location into a set of the non-zero pixels in the reference mask. We evaluated several different sets of pixels used to represent each plot mark in the mask. Section 3 describes our attempt to optimize the localization of point objects on the plots in our data using different size labelled masks (3.1) and then to optimize the classification of different geometric objects within the plots in addition to finding the points (3.2).

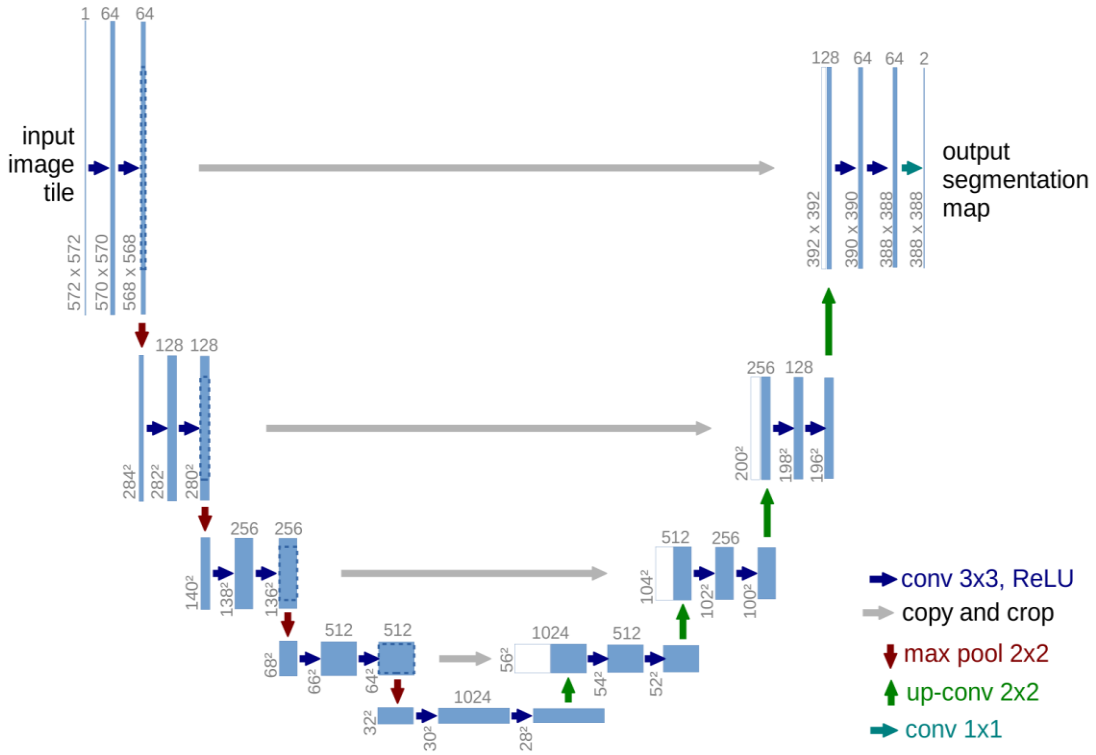

Figure 1. U-Net model architecture showing the different convolutional layers (blue arrows) and their respective levels. Each blue box is a multi-channel feature map with the channel count denoted on top of the box and the spatial dimension at the lower left edge

3. OBJECT LOCATION AND CLASSIFICATION MODELS

3.1. Plot point locations

The most frequent geometry contained in our data set is an open circle, and we have many plots containing only this one type of icon. To first look just at locating plot points, separate from classifying objects by geometries, we collected a set of these plots and annotated the training data to see how accurately we can locate the open circles on the plots. The center of each circle was marked on the training data by a small circle that varied in diameter. We used a set of 30 of these plots, breaking them into overlapping sub-images for training and validation, and an additional 20 plots for testing. The fact that each image (and its accompanying spreadsheet of data points) had to be individually edited and checked limited the number of images we included in our set. We performed some data cleaning on these raw images: using the brush functionality in ImageJ [11], we removed any axis numbers or other written information that looked like any of the geometric objects. All that remained on the plots were the plot axes, plot points, and any lines or curves in the plot that represented interpolation of the data. The data is augmented for training by flipping each sub-image that contains plot points first horizontally and then vertically. (Axes markers will be recognized as objects in the final output of our model.)

Figure 2 shows an example of marking the training masks and the resulting inferenced mask marks. In the figure, the outcome of the network is overlaid onto the plot. The outcome of the network resembles the labeling of the training data: there are clusters of pixels of the same approximate size on the input mask. Test location points are found by collecting the overall mask

from network outcomes, finding the clusters of marked pixels in the outcomes, and finding the centroid point of each cluster, using Python image processing libraries. Pixel clusters less than 5 pixels are eliminated. The remaining center point locations are tested against the original manually collected list. Table 1 shows our results. Using a smaller mask marking helps to localize the points, but too small a marking results in a model that cannot find all the plot points. The 5-pixel radius circle marking gave the best overall results.

Table 1. Two-class model results for 275 plot points from 30 different plots. Columns correspond with numbers of objects found or not found by the model, and the sample mean and standard deviation (mean dist. and std dist.) of the distances in pixels from the manually labeled locations.

Mask Radius, pixels	Correctly identified	Not found	Mean dist.	Stdv dist.
2	232	43	2.05	0.83
5	262	13	2.07	1.01
10	226	49	2.89	1.04

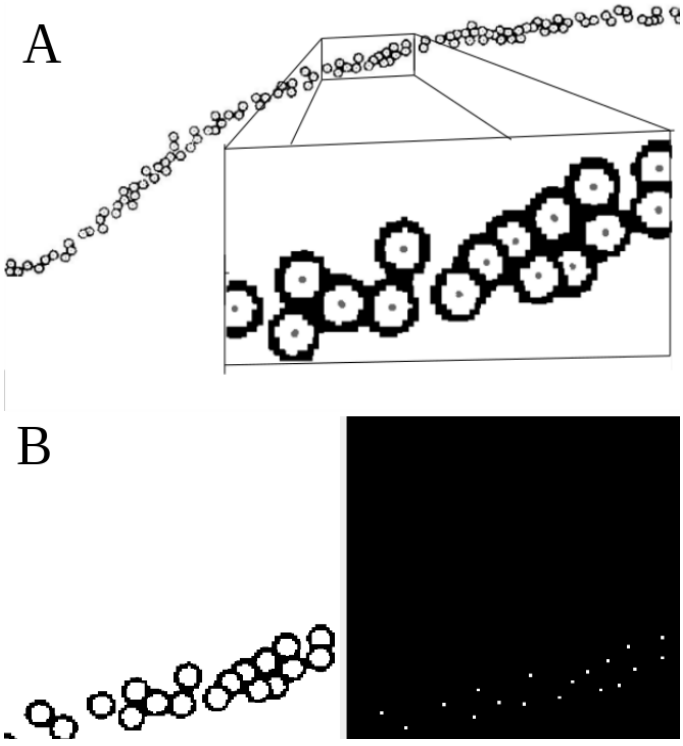

Figure 2. a: Outcome of the two-class model shown by overlaying the output masks on the original plot. Taken from [12]. The insert shows a closeup of a portion of the center of the plot (marked on the plot). The dots in the center of each circle display the outcome of the model. b: Example of training data: left: a 512x512 section of the plot from [12]; right: the corresponding annotation for network training.

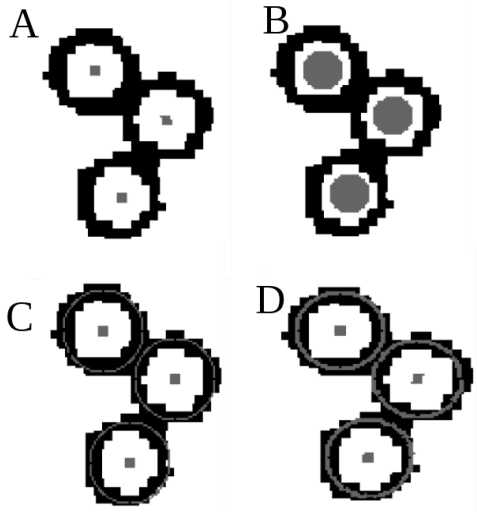

Figure 3. Four different types of segmentation masks used: A.) small center marks, B.) larger center marks, C.) small center mark with a 1-pixel outline of the object, D.) small center mark with a 2-pixels outline of the object. All markings are shown in gray, with circles shown in black.

3.2. Geometric Object Classification

3.2.1. Three-class model: open and filled circles

The segmentation masks that work well for locating plot points result in large classification errors when we extend our model to several different geometric objects. As the simplest case we show a three-class model, open circles, filled circles, and background. We selected 45 plots that contained either open circles, filled circles, or both. Of the 45 plots, 14 had only open circles, 12 had only filled circles, and 19 had both. The mean height of the images was 3133 and the mean width 3420. The mean number of plot points per image was 76.4. They were edited as before to exclude axis numbers resembling circles and divided into overlapping 512x512 sub-images. In most cases, the fraction of sub-images with no geometric objects (plot points) is more than half. The total number of training and validation sub-images was 6033, 80% of which were used for training. Training masks were created by placing a circle of radius 5 (optimal size for point location) at the known positions on the images. In a sample test image with 107 circles, 30 filled and 77 open, the location of the points was found with high accuracy (mean error 1.15 pixels off), but the classification errors were high: 16 of the 77 open circles were classified as closed circles. All of the filled circles were classified correctly.

We designed a set of different segmentation markings to create the training data, in order to see how much more, or what different information was needed for our network to classify the different circles. Figure 3 shows some of the masks that we used. In terms of post-processing, a simpler mask is preferable if that can achieve our goals. Smaller center markings are easier to separate in the inferenced images. But to provide more information to the network about the location of each object, we created more extensive markings, with larger center marks or outlines of each geometry with different thicknesses, 1 pixel and 2 pixels. A very thin outline could be eroded away in the inferenced image and might be helpful for supplying extra information to the network to differentiate between the geometric objects. When post-processing inferenced images, using annotations with the mask B, we included a method to erode the outlines and to find multiple center points when those larger center marks overlapped.

Using the same 45 plots as above, we created four different classification models to classify open vs. filled circles, with the 4 different types of masks shown in Figure 3. In general, smaller masks, both smaller for the central radial mark and thinner circular outline, led to higher correct classification levels; however, the addition of the outer marking did not change the classification rates in separating open from filled circles. The larger segmentation masks were not needed to distinguish between open and closed geometries and produced reduced accuracy in finding the location points. Table 2 shows outcomes from three sample test images: one with open circles only, one with closed circles only, and one with open and filled circles together. The test images are pictured together in Figure 4.

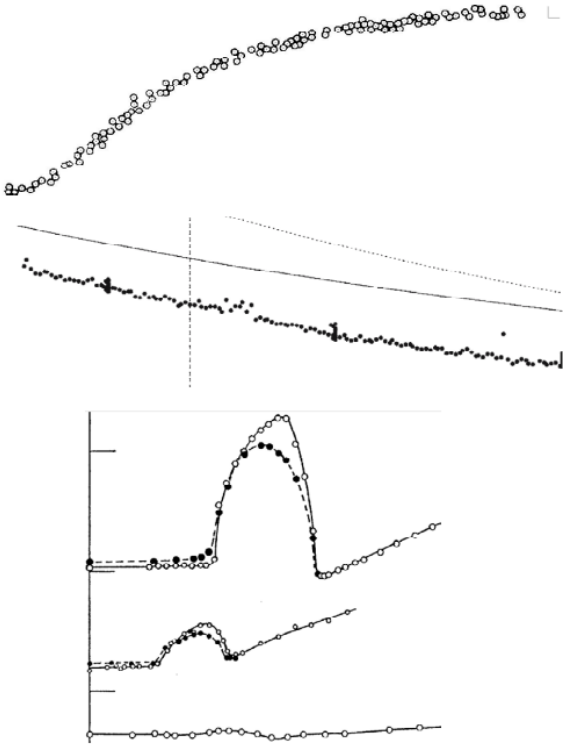

Figure 4.Test images for open vs. filled circle classifications. Top: open circles only [12], Middle: filled circles only [13], Bottom: both open and filled circles [14].

The first test image is the same image as in Figure 2 with 112 open circles. All circles found in the image were classified correctly as open circles, except in the case of the larger center markings (5-pixel radius, mask B in Figure 3), where 52 of the 78 open circles were incorrectly classified as filled circles. The second test image contains 121 filled circles. The classification worked well (all circles that were found were classified correctly). The third image contains 78 open and 30 filled circles. The network trained with larger mask marks (B in Figure 3) had a large number of open circles that were classified as closed circles (52 of the 78). All of the other models had correct classifications of open and filled circles. Smaller markings were essential to classify the open circles, with little difference in the size or presence of the outer marking. It was only the open circles that did not classify correctly; the varied shape of the inner white space makes those objects more complex than the filled circles.

Table 2. Results from classification of open/filled circles with four different segmentation masks: A=small dots, B=larger dots, C=small dots 1-pixel outline, D=small dots 2-pixel outline (as shown Figure 4). Shown are number of objects correctly and incorrectly classified, the number of objects not found by the model, and the sample mean and standard deviation (mean dist. and stdv dist.) of the distances in pixels from the manual locations. For the third image, we also show number of false positives, which did not occur on the other images.

a. Inferencing on image with 112 open circles.

	Correct	Incorrect	Not found	Mean dist.	Stdv dist.
A	109	0	3	1.32	0.68
B	34	77	1	1.40	0.91
C	109	0	6	1.84	2.20
D	106	0	3	1.35	1.16

b. Inferencing on image with 121 filled circles.

	Correct	Incorrect	Not found	Mean dist.	Stdv dist.
A	117	0	4	1.71	1.43
B	117	0	4	2.41	2.12
C	117	0	4	1.95	1.56
D	114	0	7	2.22	2.29

c. Inferencing on image with 78 open and 30 filled circles

	Correct	Incorrect	Not found	False pos.	Mean dist.	Stdv. dist.
A	107	0	1	2	1.19	0.85
B	55	52	1	0	1.63	1.93
C	105	2	1	1	3.55	0.54
D	107	0	1	1	7.65	2.45

3.2.2. Three-class model: open circles and open squares

To classify objects of different shape, we started with the example of open circles and open squares. The training set for this group is smaller, as there are fewer examples. We used 14 images of plots with circles and squares, which produced 2676 sub-images and ran four models using the four different types of masks. Here the results varied depending upon which mask was used for training: more information in the mask (using outer boundaries) led to better object classification. We show here our best and worst results, one with very sharp objects in the images for which the classifications were good and one with circles and squares that are more ambiguous, which led to less accurate classification. Figure 5 shows both plots [15][16].

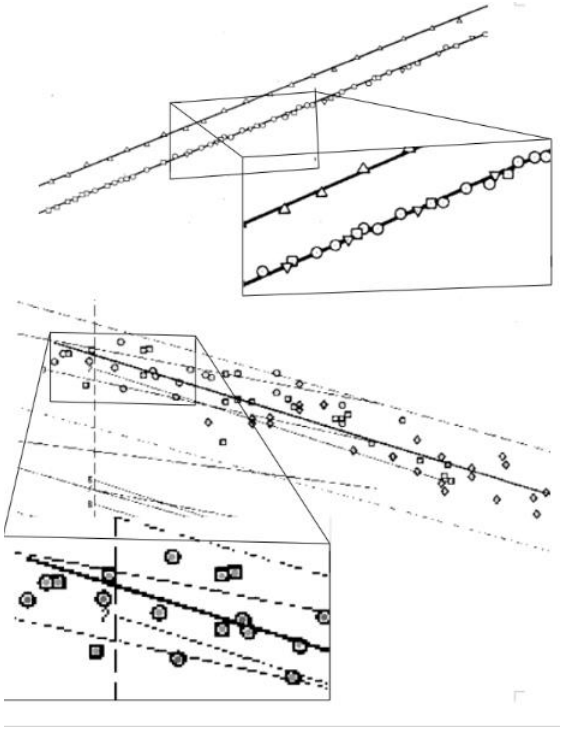

Figure 5. Two different images of plots used to test our circle/square classifications. Below each plot is a magnified section of each that shows the ambiguity of some of the circles and squares in the second plot

Table 3a displays the results of inferencing the models on the top image in Figure 5, which contains 43 circles and squares. Overall, different segmentation masks lead to the best point localization vs. the best object classification. For object location, the best model is again made with the smallest marking on the masks. For classification, however, the best model has a small marking at the object center but includes an outer marking in the shape of either the circle or the square. Mask A, with the very small 2-pixel radius marking, has the highest error rate in distinguishing between the circles and the squares, even in a very clear plot image, although it was useful to find the point locations. A combination of models is needed here to achieve both optimal point localization and object classification.

Table 3. Results from classification of open circles and open squares, with four different segmentation masks: A=small dots, B=larger dots, C=small dots 1-pixel outline, D=small dots 2-pixel outline (as shown Figure 4). Shown are number of objects correctly and incorrectly classified, the number of objects not found by the model, and sample mean and standard deviation (mean dist. and stdv dist.) of the distances in pixels from the manual locations.

a. Results from best image, top of Figure 6

	Correct	Incorrect	Missing	Mean dist.	Stdv. dist.
A	26	16	1	1.98	1.03
B	43	0	0	2.05	1.77
C	43	0	0	3.50	4.18
D	42	1	0	6.01	5.13

b. Results from worst image, bottom of Figure 6

	Correct	Incorrect	Missing	Mean dist.	Stdv. dist.
A	41	10	18	2.43	1.17
B	44	11	14	2.28	1.01
C	42	13	14	4.97	5.36
D	59	6	4	5.47	5.31

3.2.3. Four-class model: open circles, open squares, open triangles

Using each of the types of segmentation masks in Figure 3, we trained multi-class U-Net models with four classes: open circles, triangles, squares, and background. We took 32 different plot images containing open circles, open triangles, and open squares, and created a set of training and validation data. It contained 6615 512x512 sub-images, 80% or 5292 used for training. Of the 32 plots, 11 contained circles, triangles, and squares, 18 only 2 of the 3 geometries, and 2 plots contained only squares, which were the least frequent in the other plots. The mean height of the images was 3737 and the mean width 4164. The mean number of plot points per image was 48.8.

The results show that our more complicated segmentation masks greatly increased the accuracy of our object classification, although the accuracy of locating the plot points decreased somewhat. The output was tested against several test images containing all three types of geometric objects, with groups of overlapping objects in each test image. An example test image is shown in Figure 6, taken from [17]. It contains 27 circles, 43 triangles, and 36 squares. Using our original marking on the segmentation masks (the radius=2 circles), only 70 of the 106 objects were successfully classified, 14 were not found in the output, 20 were classified incorrectly (see Table 4). However, for each object that was located, the location was found precisely; the average location error was 2.37 pixels, with a standard deviation of 1.61.

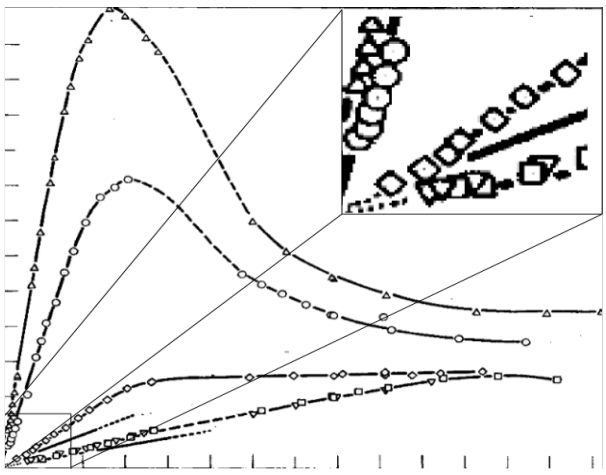

Figure 6. Test images for four-class model; taken from [17]. The upper right corner shows a magnification of the bottom left corner of the plot as marked, displaying some of the overlapping geometric objects.

Enlarging the central mark of each object in the masks of the training set decreased the number of objects not found at all by the models. Adding an additional outer circle, triangle, or square increased that number correctly classified. The results of the classifications from all 4 models, derived from the four different types of masks, inferencing the image in Figure 4 are also given in Table 4. Some of the other test images we used gave better results than those in Table 4, but the general trends were the same: the original masks with the small dots consistently had numerous

objects that were not found and incorrect classifications. The number of objects not found decreased consistently using any of the masks except the original one, and the addition of the outline in the segmentation masks increased the number of correct classifications (although they made the post-processing a little harder). Errors in locating the centers of the geometric objects, however, were lower with masks that did not contain the outlines of the geometries. The insert of the magnified plot in Figure 4 shows examples of the overlapping geometric objects and the small deviations from pure geometric shapes.

Table 4. Four-class model results for 106 objects on a plot from [17]. Columns correspond with numbers of objects correctly and incorrectly classified, numbers not found by the model, and the sample mean and standard deviation (mean dist. and stdv dist.) of the distances in pixels from the manual locations.

mask	Correctly classified	Incorrectly Classified	Not found	Mean dist.	Stdv dist.
Small dot	70	20	16	2.37	1.61
Larger dot	88	13	5	2.60	2.32
thin outline	82	18	6	4.95	4.49
thicker outline	94	8	4	5.76	4.81

Because the training sets with different types of masks led to different outcomes in the inferenced images, we can use the best aspects of several of the resulting models to get the best classification and smallest errors in locating the points of the plots. Use of the thicker outlines for the circles, triangles, and squares, in general led to the highest rates of correct classification, but also the highest location errors. The best results were found by using the models including segmentation outlines for the classification, in conjunction with the outcomes of the models without the outlines to find the correct locations; a good estimate for the locations and good classifications are both achieved.

4. CONCLUSIONS

We experimented with a U-Net network to find the locations of geometric objects in plot images. Our goal was to locate the plot points within 5 pixels of the ground truth data, and were successful locating objects with this accuracy. The model was sufficient to detect the variety of geometric objects in our collection, but not sufficient in every case to distinguish between some of our geometries. Putting more information into the image masks that the network uses for learning improved the performance of our classification of different types of geometric objects. The more complicated masks led to less accurate point locations on the plots, however. Distinguishing between filled and open versions of the same type of object, filled vs. open circles for example, did not benefit from marking the outer boundaries of the circles in the training masks. Filled circles had a much higher correct classification rate than open circles, since they do not have the ambiguity of an inner circle shape.

We had hoped that the classification of objects in these plots would have a very high accuracy, which we only see when inferencing plots whose plot points are represented by clear circles, triangles, and squares, whose shapes are not at all ambiguous due to the clarity of the image. There is quite a bit of variation in the shapes of individual circles, triangles, and squares. As usual with these models, the more we understood about the variation in our training data, the better the model we were able to produce. Perhaps because of the simple geometry of the objects and the two-toned gray scale of the images, we found that our localization was as accurate as was required without the use of loss functions previously used to add more weight to the background of images or to classes hard to classify.

REFERENCES

[1] . R. Girshick, J. Donahue, T. Darrell, & J. Malik, (2014) "Rich feature hierarchies for accurate object detection and semantic segmentation", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp580-587.

[2] S. Ren, K. He, R. Girshick, & J. Sun. Faster R-CNN, (2015) "Towards real-time object detection with region proposal networks", Advances in Neural Information Processing Systems 28. Curran Associates, Inc., pp91-99.

[3] J. Redmon, S. Divvala, R., & A. Farhadi, (2015) "You Only Look Once: Unified, real-time object detection", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp779-788.

[4] J. Redmon & A. Farhadi, (2018) "YOLOv3: an incremental improvement", arXiv preprint arXiv:1804.02767.

[5] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, & S. Reed, (2016), "SSD: Single shot multibox detector", European Conference on Computer Vision (ECCV), pp21-37, 2016.

[6] E Goldman, R. Herzig, A. Eisenschtat, J. Goldberger & T. Hassner (2019), "Precise detection in densely packed scenes", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp5227-5236, 2019.

[7] O. Ronneberger, P. Fischer & T. Brox, (2015) "U-Net: Convolutional networks for biomedical image segmentation", Medical Image Computing and Computer-Assisted Intervention (MICCAI), pp234-241.

[8] J. Ribera, D. Guera, Y. Chen & E. Delp, (2019) "Locating objects without bounding boxes"' IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp6479-6489.

[9] A. Kirillov, H. Kaiming, R. Girshick, C. Rother & P. Dollar, (2019) "Panoptic segmentation", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp9404-9413.

[10] T. Lin, P. Goyal, R. Girshick, K. He & P.Dollar, 92017) "Focal loss for dense object detection", IEEE International Conference on Computer Vision (ICCV), pp2980-2988.

[11] C. Schneider, W. Rasband & K. Eliceiri, (2012) "NIH Image to ImageJ: 25 years of image analysis", Nature methods, Vol.9, No. 7, pp671-675.

[12] S. Novikova, (1960) "Thermal expansion of germanium at low temperatures", Soviet Physics - Solid State, Vo.l 2, pp37-38.

[13] M. Kehr, W. Hoyer & I. Egry. A new high-temperature oscillating cup viscometer. Intl. J. Thermophysics, Vol. 28, pp1017-1025, 2007.

[14] D. Detwiler & H. Fairbank, (1952) "The thermal resistivity of superconductors", Phys. Rev. Vol. 88, pp1049-1052.

[15] J. Rayne, (1956) "The heat capacity of copper below 4.2K", Australian J. Physics.Vol. 9, pp189-917.

[16] K. Aziz, A. Schmon & G. Pottlacher, (2015) "Measurement of surface tension of liquid nickel by the osciallating drop technique. High Temperatures – High Pressures", Vol. 44, pp475-481.

[17] G. White & S. Woods, (1955) "Thermal and electrical conductivities of solids at low temperatures", Canadian J. Physics, Vol. 33, pp58-73.