Investigating Abstracted Visual Concepts Learning with Deep Convolutional Neural Networks and Alternatives

Keen You

May 11, 2022

1 Introduction

Deep neural networks are undoubtedly successful at many computer vision tasks. However, as mentioned in class, deep neural networks struggle with tasks that require learning abstracted visual concepts, such as the same-different relations. Figure 1 (1) displays a few examples of such task. A human judgement for such tasks should be consistent regardless of the positions of the shape. However, for a deep CNN, the prediction may be different if we change the positions of the shapes as the generated feature representation will be different.

Figure 2 2 is an even more extreme example. Human judgement will be unaffected by the background, but deep CNN will be significantly affected as the images are very different, even though the abstracted visual concepts and task (find shapes and compare) are identical.

This observation raises the question: what exactly are deep CNNs learning? Even though CNNs are inspired by the human vision system, it seems that humans and deep neural networks are learning different things: humans learn the underlying abstract visual concepts whereas deep neural networks learn a set of rich and complex features that are position and dataset-specific.

In this project, I want to investigate what deep CNNs are learning for tasks that require understanding of abstracted visual concepts and what alternative approaches there are. Particularly, I want to further explore various counting star related tasks in homework 4 q6.

2 Counting Stars with CNN

In this section, we explore the task: given image, predict the number of stars in the image. In homework 4, a CNN achieves training performance of 0.1381 mse and testing performance of 0.1405. Accuracies are 0.89 and 0.82 respectively. Visualization of the filters did not really provide much insights on what fetures the CNN is learning. Thus, I suspect that the CNN is relying on some other signals, such as the number of white pixels. I will verify the hypothesis now by 1) examining the number of white pixels for each class; 2) evaluating the plausibility of linear models; 3) CNN on zoomed in images (such that number of white pixels are similar for each class).

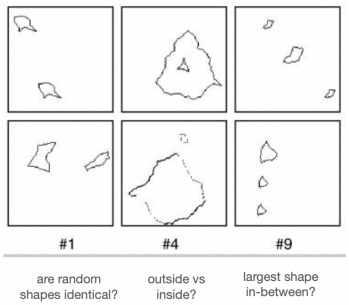

Figure 1: Same-difference visual task.

Figure 2: Extreme same-difference visual task.

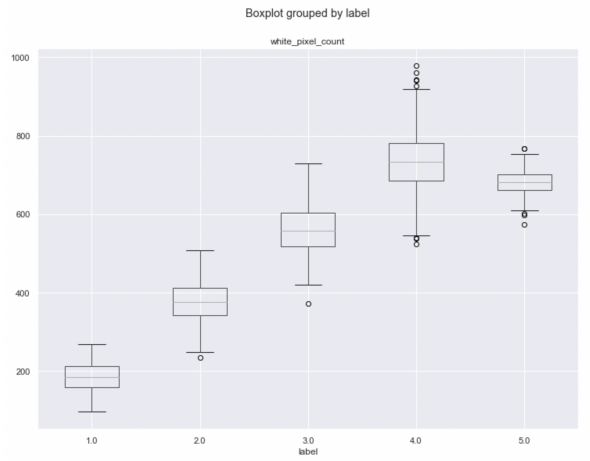

Figure 3: The number of white pixels in each discretized image grouped by class labels.

With figure 3 (3), it seems like there is noticeable difference in the number of white pixels for different classes. Figure 4 (4) displays a scatterplot of number of white pixels in discretized image colored by class label. The training set and testing set exhibit the same trend. From these observations, we see that the size of each star in images with 5 stars is smaller compared to that for other classes. This results in smaller number of white pixels for class 5 compared to class 4. This destroys the linear trend from class 1 through 4. Thus, a linear classifier may not suffice. The data that a linear model needs to be fitted to is displayed in figure 5 (5). A linear model is fitted on the dataset to verify this claim. Particularly, a linear regression model is trained on bag-of-features (flattened image as a vector representation for each image). The training accuracy is 0.79 and the testing accuracy is 0.57.

The performances by linear model are significantly worse compared to CNN, which is expected. However, the performance is far better than random (20%). This suggests that without any concept of the shape star, the classifier is already doing decent at this task, let alone incorporating other superficial features and even features that are not observable for human eyes. In addition to number of white pixels, other very likely features that CNN can make use of include positions of black pixels. For example, for images with 1 star, roughly three quadrants of the image are black, and none other class exhibitis this property. In general, for this task, there are many computational features that could be used to make predictions, rather than actually learning what a star shape is.

3 Modified Counting Stars with CNN

For humans, a star shape is a star shape regardless of its size ie. translates to the same abstract visual concept. However, this will induce different feature representations in CNN. Is it going to affect predictions? We will find out using a modified version of the counting stars task. One crude way of doing this is to randomly zoom into some of the images, such that stars belonging to the same class do not always have the same size (number of white pixels). It is possible that part of some stars lie outside the images, but for a human, it should not affect judgement as one can still count the number of stars (if assuming all objects are stars). An example of zoomed in image is shown in Figure 6 (6). 2000 images are randomly selected from each class to be zoomed in. Using a CNN with the same

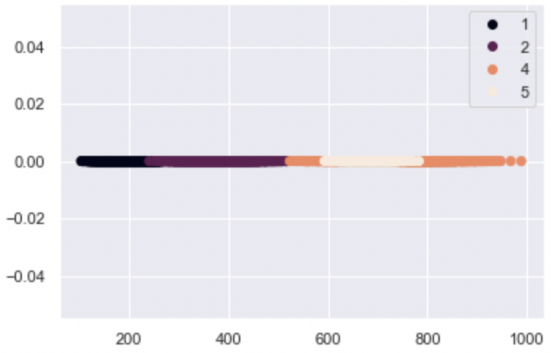

Figure 4: The number of white pixels in each discretized image colored by class labels.

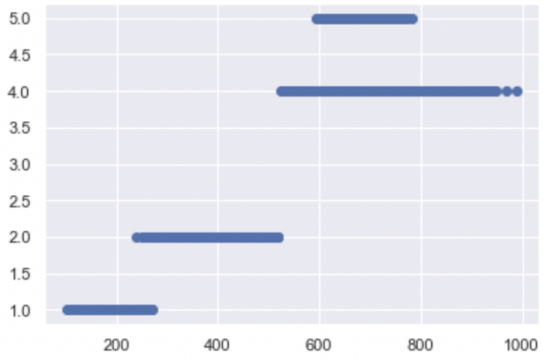

Figure 5: Training dataset.

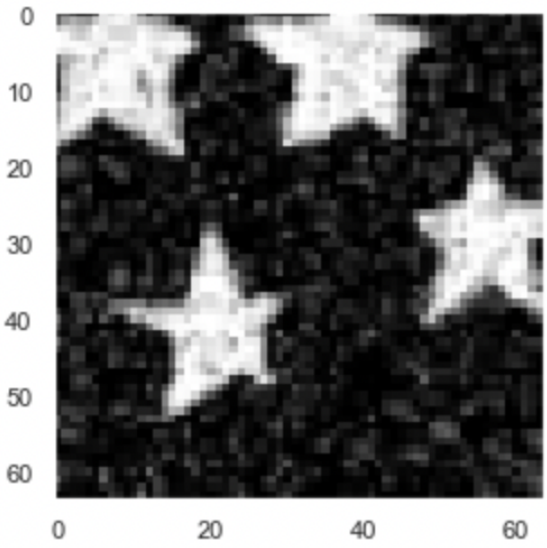

Figure 6: An example of zoomed in image. We can tell that the number of stars in this image is 4, even though parts of the top half stars lie outside the image.

architecture as before results in a training accuracy of 0.25 and test accuracy 0.19. The performance on this dataset is significantly worse compare to original dataset. Performance on the original dataset is 0.89 on train and 0.82 on test. This is expected from a computational perspective, but not expected from a abstract concept learning perspective. If someone gets the abstract concept of a star shape, or just a shape in general, it is expected that they are able to count the number of shapes even when images are zoomed in.

4 Counting Star Tips

In this section, we explore the tasks: given image, predict the number of tips that the single "pointy star-like object" in the image has, and given image, predict the number of stars that have exactly 5 tips. These tasks require specific understanding of the concept of a tip, which is a subpart of the concept of a star.

The CNN model from homework 4 does a decent job on the first task with a 0.96 training accuracy and a 0.92 testing accuracy. Some inspections of filters from different filter layers do not provide much insights on what exactly the CNN is learning. Even in the last layer of filters, it does not seem like they are directly learning a template/the concept of a tip. In contrast to a decent performance in first task, even CNN with more parameters does a terrible job (20% accuracy on the test set) on the second task.

Similar to previous part, the number of white pixels of discretized images are examined, displayed in Figure 7 (7) and 8 (8) respectively. For the first task, the number of white pixels in each image can be an important signal for its label. The number of white pixels for the second task are a lot more similar compared to previous dataset. Thus, this may not be a useful feature for CNN in classification.

5 Learning the Concept of A Star Tip

In previous section, we see that CNNs are highly likely to rely on other features in the tasks of counting number of tips and counting number of stars with 5 tips rather than learning what a tip is. In this section, we will explore learning the concept of a star tip directly, which is framed in two different ways: 1) corner detection, and 2) ridge detection. We will then analyze the results qualitatively.

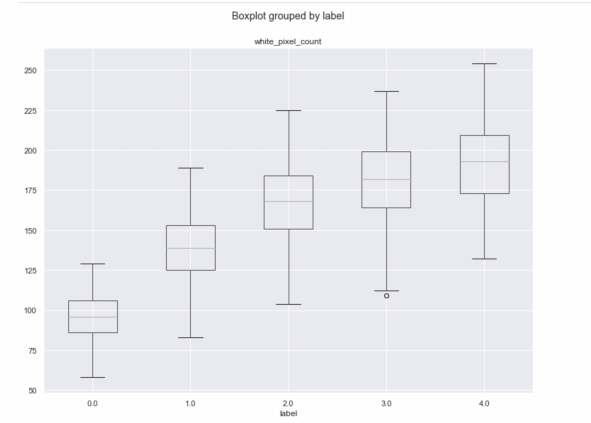

Figure 7: White pixel distributions in second dataset.

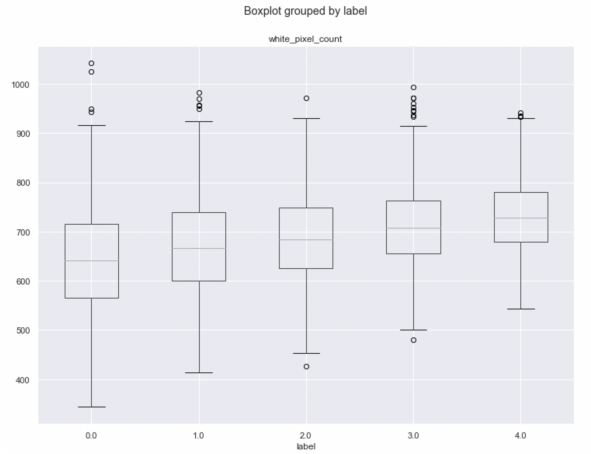

Figure 8: White pixel distributions in third dataset.

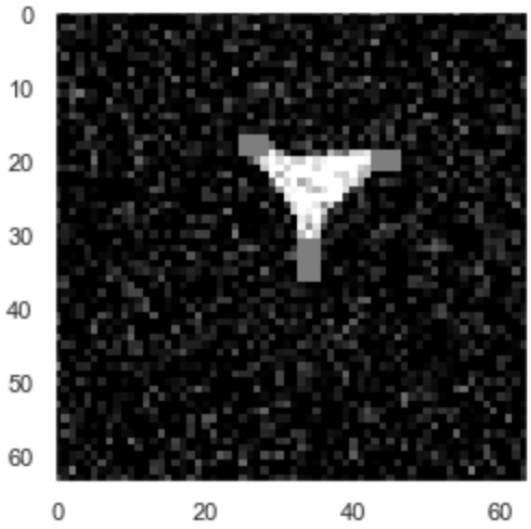

Figure 9: Corner detection on original image. Corners labeled in gray.

5.1 Corner Detection

In the corner detection approach, we apply Harris Corner Detection on original image or thresholded image. Some examples are shown in Figure 9 (9), 10(10), and 11(11). We see that the detected corners are quite noisy, especially for stars with more tips. The main challenge is to pick a threshold to determine what is considered a corner. If the labeled corner is too big with a small threshold, they may fuse together for stars with more tips. On the other hand, with a larger threshold some smaller tips may not be detected.

5.2 Ridge Detection

In the ridge detection approach, we filter an image with the Frangi vesselness filter, either to detect white ridge or black ridge. Examples are shown in Figure 12(12) and 13(13). We then discretize the detected output and use the number of white areas as a prediction for the number of tips. The predictions for the 3-tip example are shown in Figure 14(14) and 15(15) respectively. Refer to jupyter notebook output for more outputs.

5.3 Discussion

Finally, both approaches are applied an example from the third dataset. From the results, we can see that detecting black ridges is disadvantageous here because the stars are more compact in this dataset. Overall, detecting corners directly seem better qualitatively. Detecting white ridges is likely to merge the ridges together at the center of the star and detecting black ridges is likely to merge among the starts if multiple stars are present in the image. However, detecting corners is not doing wonderfully as it is difficult to accomodate tips of different angle from stars with different number of tips.

5.4 Conclusion

In conclusion, CNNs do not learn the concept of abstracted visual concepts while training on classification tasks. Thus, we explore the direct learning of an abstracted concept, star tip, using corner detection approach and ridge detection approach. While the results are intuitive and capture the labels to some extend, in order to make it a mature algorithm that produces good performances, more work needs to be done to refine the algorithms and tune relevant parameters.

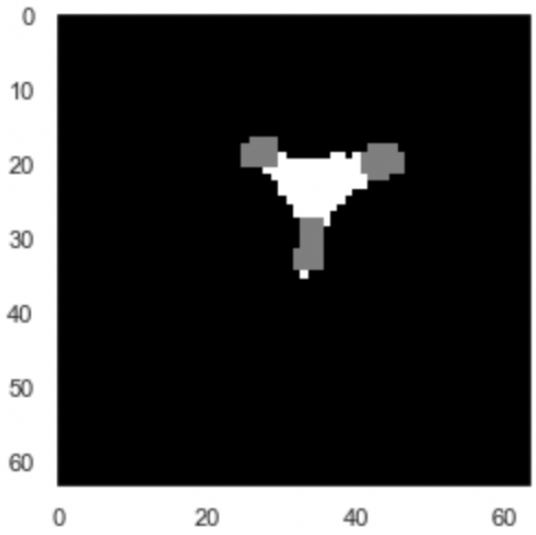

Figure 10: Corner detection on thresholded image. Corners labeled in gray.

Figure 11: Corner detection on original image with 7 tips. Corners labeled in gray.

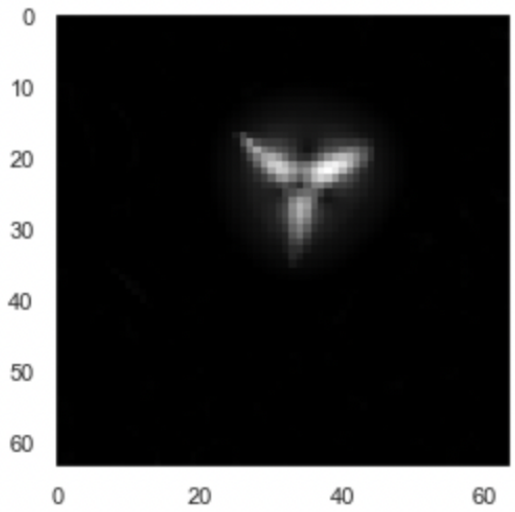

Figure 12: Detect white ridge

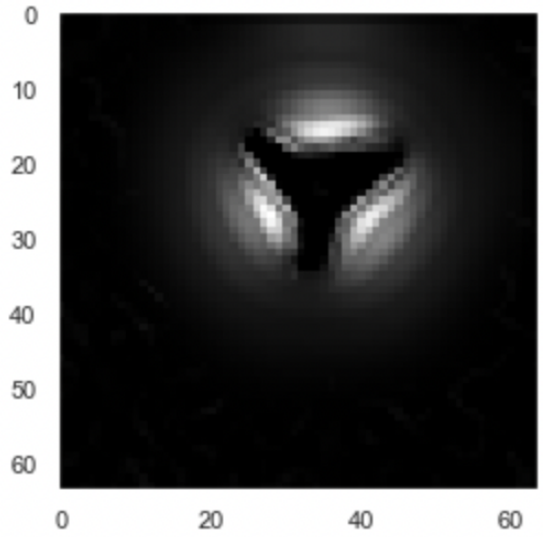

Figure 13: Detect black ridge

Figure 14: Prediction using white ridge detection.

Figure 15: Prediction using black ridge detection.