Geometric Classification Project

Lee A. Barnett and Calvin Deutschbein

Abstract— **With the introduction of deep learning on convolutional neural networks applied to image categorization, it has become important to learn the nature and capabilities of these networks more precisely. However, traditional applications for convolutional neural networks often have a lack of clarity in what constitutes ground truth. To work towards resolving this, we have a developed a dataset of rendered black triangles centered on a white background and trained a convolutional neural network to look for objective mathematical properties of the triangles. Furthermore, to try to gain greater understanding on the nature of the neural network, we also exposed the trained neural network to a set of Extension Tests, a set of more diverse data including other polygons, triangles with other gray-scale colorings, cropped triangles, and multiple triangles. We found that the convolutional neural network had some interesting irregularities in labeling but overall did relatively well especially given the difficulty of some of the edge cases between various categories.**

I. Introduction

Deep learning using convolutional neural networks has had a large and rapid impact on the field of computer vision, especially in visual classification. However, convolutional neural networks are immensely complicated, often having millions of parameters, and are not as comprehensively understood as some other computer vision methodologies as a result of this and their relatively recent entry into use. Convolutional neural networks have rapidly approached (and in some tasks, exceeded) human performance on a variety of tasks, but visual recognition and classification often suffer from a lack of clarity in ground truth. While it is perhaps unsurprising that neural networks, which in many ways closely mimic human biological neural structures, produce similar results to humans performing the same tasks, this is a far cry from accessing what actual computation (if any) is occurring in the context of a neural network.

There appears to be very little prior work in the application of a convolutional neural network to tasks with objective ground truth, assessing the network's performance, and attempting to use the network in an extensible way to gain further insight on its internal representations. To that end, the main contributions of this paper are as follows:

- We have generated a dataset in which all categories are precise, unambiguous mathematical definitions of the objects shown.
- We have trained a neural network to perform classification on this dataset and have evaluated its performance at classification.
- We have applied the trained neural network to a set of Extension Tests containing data very different from that in the training set in order to attempt to discern some of the mechanisms by which classification occurs in this context.

II. Geometric Classification

Planar geometry is used to describe the principles underlying shape, size, and the properties of two-dimensional space. This study has an ancient history, having practical and direct applications to various spatial tasks including astronomy, navigation, construction, and surveying. Basic geometric shapes, such as triangles, circles, angles, and ellipses, are generalized forms of those that occur naturally, idealized in order to study spatial relationships from an abstract point of view.

Natural images capture two-dimensional projections of our three-dimensional space, and so the notions of spatial relationship distilled by planar geometry apply to the basic forms appearing in natural images. For example, in figure 1 below, the notions of symmetry and angle appear clearly as remarkable properties of the spatial relationships in the image.

Fig. 1. An orange *Gazania* flower in full bloom [5].

The task of classifying shapes according to their basic geometric properties is straightforward: for example, a triangle is isosceles if it has two sides of equal length (in this paper, isosceles means exactly two sides of equal length), a parallelogram made of right angles is a rectangle, etc. From these basic classification tasks it is not difficult to expand to slightly more complicated ones. For example the two triangles pictured in figure 2 are congruent; this can be tested by measuring that two pairs of equal-length sides are separated by equal angles, as seen in this image, or by other similar tests, such as measuring that two triangles have three pairs of equal-length sides.

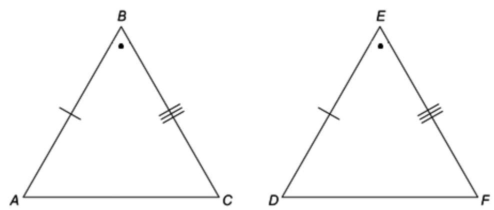

Fig. 2. Two congruent triangles.

As can be seen from this example, there is a degree of compositionality to much of planar geometry, in that problems which seem more complicated can be solved by inspecting the same properties used to solve simpler problem instances. As another example, deciding if a quadrilateral is a rhombus can be done by deciding if it is composed of two congruent, isosceles triangles. However, it is not always immediately clear which basic geometric properties are at play in a problem, and how they are used; e.g., the convex hull decision problem asks whether, given a set S of points on the plane and a specified point P, is P a member of the convex hull of S (the smallest convex region boundary on the plane containing S)? Many geometric classification problems exist, with varying degrees of difficulty.

III. Deep Learning

Image categorization has increasingly moved toward using deep learning methods of many-layered convolutional neural networks (or CNNs) [6]. These networks are the product of many parameters, often run on specialized hardware such as GPUs, and trained on enormous amounts of data.

These methods have demonstrated significantly lower error rates in categorization than methods that do not use deep learning and are now competitive with human categorization [1]. However, due to a lack of theoretical research their precise nature is not fully understood, and as a result deep learning and CNNs are often treated as "black-box" methods of computation. This has led to a lack of clarity about how to approach certain weaknesses of these methods [2][4].

Our goal is to train a network using images of geometric shapes to perform simple geometric classification tasks described specifically in section IV.B. The purpose is that in training the network according to simple properties, the extracted features will represent exactly these properties. After training the network on triangle properties, we use the network to work on more complicated problems without retraining, by focusing on different classification problems that require only measuring the properties the network has already been trained on. For example, after training the network to recognize angle properties of triangles, we focus it on the corresponding properties of quadrilaterals.

IV. Implementation

To train a network for the task described, we generated a dataset of simple geometric figures labeled with the corresponding classification data. The dataset is a series of square images of polygons which for now only includes triangles. Triangles are tagged by geometric functions before being rendered according to two sets of labels: (1) as equilateral, isosceles, or scalene, and (2) as right, acute, or obtuse. Effectively two copies of the dataset were used, where the first copy is tagged according to the one label and the second copy is tagged according to the other. This was done for two purposes: first, the model we are using for training supports only single-label training out-of-the-box, and second, the classification of an image according to one label does not (and in training, should not) affect its classification according to another. Although this doubles the amount of time needed for training, since we are really training two models, we believe that multiple models will be able to give us more insight during the analysis stage of our project. While only black and white are intentionally rendered, anti-aliasing will occur along the edges of the polygon in an attempt to preserve photorealism and as an artifact of the rendering engine. Nevertheless, in all generated data images will be presented in grayscale, including in the extension tests.

A. Generating the dataset

The dataset has been generated using the image library of the LISP interpreter DrRacket. This allows creation of a dataset of any arbitrary number of elements. For the sake of this experiment, 100,000 images were generated, each displaying a centered randomized triangle in black on a white background.

Consider the case of triangle analysis. One way to generate a random instance of a triangle in a space of size max by max is by using a function for generating random points in the space:

```
(define (rand-posn)
(make-posn (random max) (random max)))
```

While DrRacket does not recommend using the function "random" for cryptography, it is nevertheless pseudo-random and more than sufficient for the purpose of dataset generation.

With the ability to generate random points, three such points may be generated together:

```
(list(rand-posn)(rand-posn)(rand-posn))
```

In the dataset's generating code, this is stored as a local variable called "3points" because at this point in the generation of an element of the dataset, processing flow proceeds in two ways. An image of a triangle is generated from "3points", and geometric properties of triangle to be generated are determined by other functions that are passed "3points" as well.

As a side note, one of the benefits of lisp is that it's straightforward nature makes seeing the equivalence between lisp code and mathematical definition relatively simple.

The image is generated using the included function "polygon" which, given a list of points and a few parameters related to render style, outputs an image of polygon of appropriate shape and size:

```
(polygon 3points "solid" "black")
```

To produce images of consistent resolution, the rendered polygons are overlayed onto a pre-rendered background in white:

```
(define bg (square res "solid" "white"))
(overlay <polygon> bg)
```

This enables triangles of different sizes and orientations to all be rendered in images of the same size. For an example, see figure 3 below.

B. Tagging

As the primary purpose of using a dataset of this type was to have an objective true ground truth, the definitions of equilateral, isosceles, right, and obtuse triangles are rigorously formulated geometrically and applied to the mathematical abstraction of the triangles rather than the rendering.

Fig. 3. An example of a rendered image. Note the anti-aliasing. Image native resolution is 256 by 256 pixels.

While inclusion of the source code that does so is perhaps beyond the scope of this paper, a few notes are probably necessary.

- Equilateral triangles are tested by applying three way equality to all three side lengths. Side lengths are determined by passing all three sets of pairs of two of the three points to a helper function that determines the distance between the points through a straightforward application of the Pythagorean theorem.
- Isosceles triangles are tested similarly but use only two way equality and iterate over the three possible combinations of sides of equal length. The same helper function is used here. Prior to testing triangles to see if they are isosceles, they are tested as equilateral. Consequently, this definition of isosceles requires exactly two sides of equal length rather than at least two sides of equal length. This was done so that each triangle would have only one side-length orientated classification.
- Scalene triangles are the triangles that are not isosceles or equilateral.
- Right triangles are tested by taking the slopes of the three lines passing through the three distinct pairs of two points. Each of the three pairs formed by taking two lines are compared against each other by testing if the reciprocal of one slope is equal to the slope of another.
- Obtuse triangles are tested by determining side lengths using the aforementioned helper functions. Side lengths are then squared. The maximum of the squares of side lengths is then tested to be greater than the sums of the other two side lengths, another straightforward application of the Pythagorean theorem.

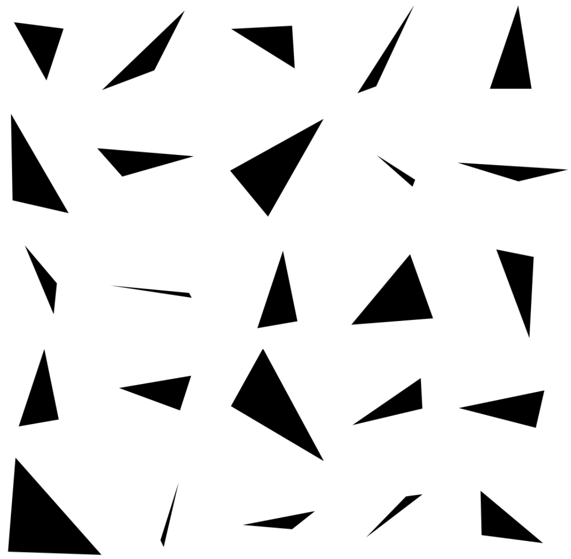

Fig. 4. A sample of some of the images from our dataset to illustrate the variety of the triangles being used.

- Acute triangles are triangles that are not right or obtuse.

Given the clear nature of code developed in DrRacket, we can have a high degree of confidence that the implementations in lisp closely align with mathematical theory. This allows the classifications to be assigned as ground truth with very high confidence.

C. The Nature of the Data

When generating triangles in this matter, triangles of different types can be substantially more or less likely to be generated. Even in the case of uniform number generation, because random points are generated rather than random types of triangles, triangles such as right triangles or equilateral triangles are extraordinarily unlikely to be generated. Consider, for example, the case of generating a third random point after two have already been selected. At most, two remaining points in the 224 by 224 size space are eligible to create an equilateral triangle, making such triangles less likely than the case in which the same point is selected for two vertices. The case of right triangles is less rare, but after generating two points there are only two lines of pixels through the space and two individual pixels sharing one coordinate each with the first two generated points in which the third point may be generated to produce a right triangle and lines of some slopes may have as few as one eligible point within the entire rendering grid.

For example, consider the case in which the uppermost and rightmost point is selected as well as the the point one below the uppermost and leftmost. In this case, there would be only two eligible points in the 224 by 224 space that produce a right triangle: the uppermost and leftmost point and the point one below the uppermost and rightmost. By contrast, in the case of simply producing an acute triangle, the entire space except the single leftmost column would be eligible and even the relatively unlikely obtuse case would be 222 times more likely than the right case (constituting all other points in the leftmost column). In terms of side lengths, there would be no eligible points to produce an equilateral triangle, only two possible points to produce an isosceles triangle (repeating either of the pre-selected points) and all remaining points would produce a scalene triangle. While a contrived example, this shows the extent to which acute, scalene triangles can easily dominate representation in the dataset.

The 100,000 image dataset contained at least 1000 images of all types except equilateral by default, so a thousand triangles were generated to be specifically equilateral.

Fully randomized triangles were considered preferable for a dataset as one of the motivations for using geometric shapes is the compositionality present natural images by triangles, and triangles in nature rarely occur as equilateral or right. Consequently, the dataset was generated as randomly as possible while having meaningful representation of all types in order to more closely mimic what can be expected to be found in the natural world.

V. Architecture and Training

To implement our project, we used the Caffe deep learning framework [7] to train two convolutional neural network on our data. Specifically, we are using the "CaffeNet" CNN model, a model based on the tutorial introduction to using Caffe. CaffeNet is a replication of the AlexNet CNN model [3] with two differences: the order of the layers is switched so that pooling precedes normalization, and the data augmentation technique that alters the intensities of the RGB channels in the training set is skipped. The purpose for using this model is first that its success is well documented, and that it comes as an "out-of-the-box" feature of Caffe, meaning that it was not too difficult to set up. The second purpose is that because the CaffeNet architecture is made freely available, it seems to be a fairly ubiquitous CNN model; the overall goal of our project is to learn

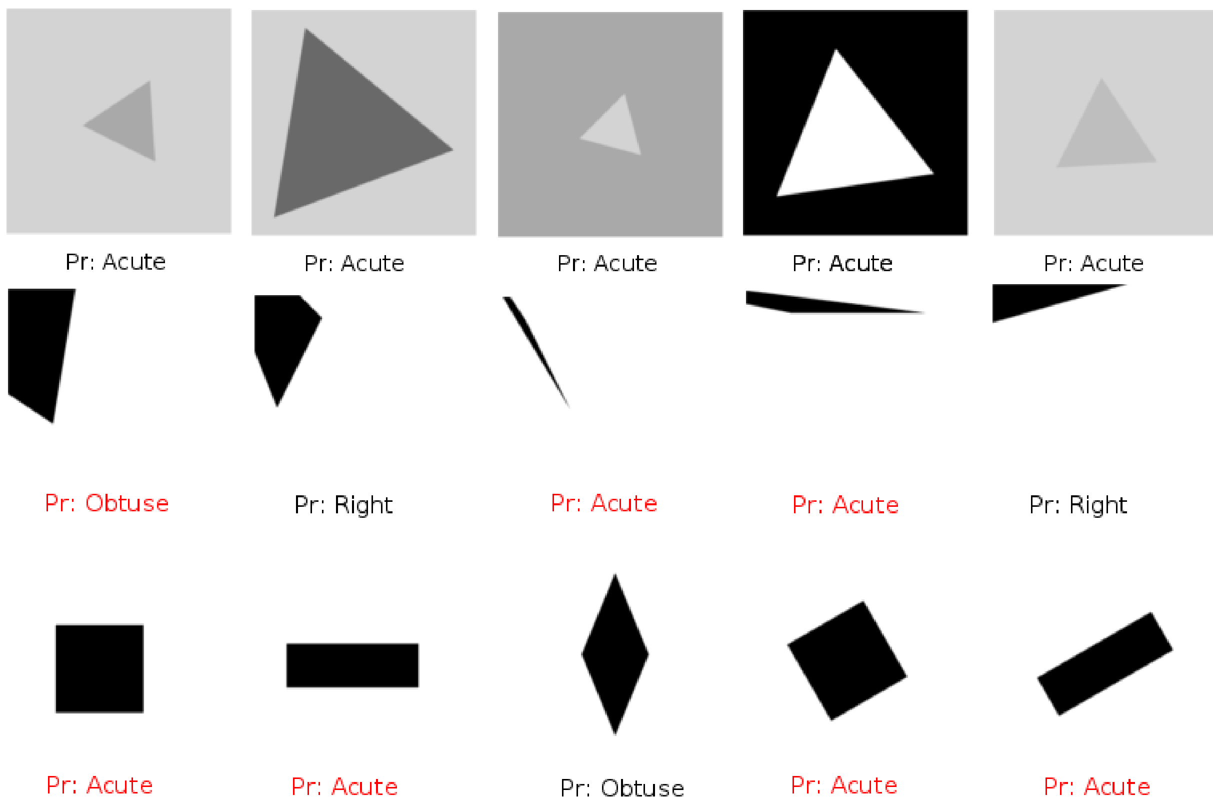

Fig. 5. Some of the Extension Tests. The predictions of the Angles network are given below each image. Labels printed in black indicate that the predicted label matched our hypothesized label, while labels printed in red indicate that the predicted label did not.

more about the nature of CNN models in general, so this seemed like a good choice.

Each model was trained using an NVIDIA GTX 770 GPU on 90,000 images from our dataset. For the "Legs" model, images were tagged according to the first set of labels described in the previous section: equilateral, isosceles, or scalene. For the "Angles" model, images were tagged according to the second set of labels: right, acute, or obtuse.

Following training, each model was evaluated on a validation set of 10,000 images from our dataset. After presenting our project in class, we re-examined our training and validation methods and found two important issues that were impacting our results.

1) The training set for the Legs network contained duplicate images that were labeled differently, causing poor performance on the validation set. The labeling is being revisited, but new results are not available to print in this paper. For this reason, the rest of the paper focuses on the Angles network.
2) The Extension Tests image results presented in class were largely incorrect. While the mean of the dataset was subtracted from the images before training, due to an oversight it was not subtracted before this classification task. This has been corrected and produced slightly more interesting results.

		Predicted Class		
		Right	Obtuse	Acute
Actual Class	Right	**39.6%**	38.3%	22.2%
	Obtuse	1.11%	**97.5%**	1.36%
	Acute	1.77%	0.22%	**98.0%**

Overall accuracy: 95.0%

Fig. 6. Confusion matrix for the Angles network. Percentages reported are per-class accuracy; e.g., 97.5% of obtuse triangles in the validation set were correctly classified as such.

VI. Results and Discussion

The overall accuracy and the confusion matrix for the Angles model is contained in figure 6 above.

A. Validation set results

While obtuse and acute triangles were very often correctly identified, right triangles were correctly identified much less often. The overall accuracy is only slightly affected by the poor performance on right triangles due to the face that right triangles made up a small proportion of the validation set as a whole. Similarly, the poor performance on right triangles could be mitigated by increasing their representation in the training set as well. As explained in the previous section, the results for the Legs model are not presented here due to mistakes that were caught too late to correct for this project.

B. Extension Tests results

The most interesting results of the Extension Tests can be seen in figure 5 above. The top row consists of images containing acute triangles colored differently from the training set images. These images were all correctly classified as acute, suggesting that the network representation of this class relies on features that are independent of coloring and shading.

The second row from the top consists of images containing partial triangles centered above and to the left of the image. The second and fifth images from the left were predicted as right triangles. This classification is considered favorable because by cropping the triangles in the top left corner, both images now contain a prominent right angle. However, all images in this row contain a complete right angle in this corner, so it is unclear why the others were not classified similarly. It is possible that this is due to the overall poor performance on right triangles, and perhaps simply by adding more right triangles to the dataset each of these images would be labeled as right triangles. It is encouraging that the fifth image in this row is labeled as right, since the image contains a full example of a right triangle.

The third row consists of images containing quadrilaterals. Our hypothesis here was that quadrilaterals containing right angles would be marked as right triangles, showing that the internal representation of the network's right triangle class depended on locating a 90° angle in the image. However, all but the center image containing a rhombus were classified as acute. It is again possible that just by increasing the proportion of right triangles in the training set, and presumably increasing the validation set performance on this class, each of these images would be classified as right. If this would not be the case, perhaps the representation of an acute triangle in the network relies on something more complicated than looking for acute angles. A distinction between acute and obtuse triangles is that, given a bounding square drawn to contain both endpoints of the largest leg of the triangle, a higher proportion of this square's area overlaps with the triangle if the triangle is acute. If the network distinguishes triangles this way, for example, then it makes sense that all of the quadrilaterals in this row were marked as acute. It is encouraging that each of these images received the same classification (in class it was incorrectly reported that the fourth image in this row was classified as obtuse; this was due to a simple error as explained in the previous section). The fact that the rhombus in the center was marked as obtuse was considered favorable because the rhombus does indeed contain obtuse angles.

VII. Conclusion

Powerful neural networks are being used without being well understood from a theoretical perspective. This is undesirable and has likely resulted in suboptimal use of these methods in spite of impressive results. By focusing an image recognizer on a very specific type of data, that is, polygons in two dimensional space, the amount of information available to the network is significantly reduced. Moreover, in this geometric context, attributes of images may be perfectly objective as a matter of geometric definition. This restriction means that if the recognizer is able to correctly identify geometric information about images, it is certainly evaluating the images for some factor that is logical equivalent to the formal, mathematical definition. We believe that our network's results on the Extension Test images suggest the representations of our image classes learned by the network, while still mathematically sound, are different than our own intuitive understanding of the differences between these types of triangles.

While this experiment was not as successful as hoped, it demonstrates the complexity of representations inside convolutional neural networks and suggests that there is more insight to be gained by studying networks this way. If future experiments can lead to more lucid and solid conclusions then it may be possible to understand CNN representations and computation from a theoretical and logical point of view.

Future work may elucidate more precisely the limits of these deep learning methodologies and provide insight into how to use them more accurately and effectively.

REFERENCES

[1] Dan C. Ciresan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. *CoRR*, abs/1202.2745, 2012.

[2] Ben Goertzel. *Artificial General Intelligence: 8th International Conference, AGI 2015, Berlin, Germany, July 22-25, 2015, Proceedings*, chapter Are There Deep Reasons Underlying the Pathologies of Today's Deep Learning Algorithms?, pages 70–79. Springer International Publishing, Cham, 2015.

[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey 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.

[4] Anh Mai Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. *CoRR*, abs/1412.1897, 2014.

[5] Sam Oth. Photo. `https://commons.wikimedia.org/wiki/File:Flower_jtca001.jpg`, 2004.

[6] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[7] Berkeley Vision and Learning Center. Caffe: a fast open framework for deep learning. `https://github.com/BVLC/caffe`, 2014.