# A Preliminary Study of Spatial Bias in Knn Distance Metrics

**Gabriel J. Ferrer**
Department of Mathematics and Computer Science
Hendrix College
1600 Washington Avenue
Conway, AR 72032
ferrer@hendrix.edu

## Abstract

A machine learning algorithm for image classification exhibits spatial bias if permuting the order of image pixels significantly alters its classification accuracy. In this paper, we explore the spatial bias of a number of different distance metrics for k-nearest-neighbor image classification. One distance metric is inspired by the convolutional kernels employed in convolutional neural networks. The other metrics are based on BRIEF descriptors, which generate bit vectors corresponding to images based on comparisons of pixel intensity values. We found that the convolutional distance metric exhibited a strong positive spatial bias, as did one of the BRIEF descriptors. Another BRIEF descriptor exhibited a negative spatial bias, and the remainder exhibited little or no spatial bias. These results lay a foundation for future work that would involve larger numbers of convolutional iterations, potentially synergized with BRIEF-style image preprocessing. The complete implementation of the work in this paper is available online at https://github.com/gjf2a/flairs33.

## Introduction

(Mitchell and Sheppard 2019) observed that Convolutional Neural Networks (CNNs) (LeCun et al. 1998) rely upon *spatial bias* for their strong classification performance. A classifier with spatial bias will perform differently if the ordering of input features is permuted. Inspired by the success of (Mitchell and Sheppard 2019) in the incorporation of spatial bias into Random Forest classifiers, this paper describes a preliminary investigation of how spatial bias can impact the error rates of k-nearest-neighbor classifiers (Peterson 2009) using different types of distance metrics.

The BRIEF descriptor (Calonder et al. 2010) is an image descriptor that has been widely employed in the robotics community due to the relatively low amount of calculation necessary to determine distances between images that have been preprocessed with this descriptor. In light of the high computational requirements of CNNs, this work seeks to determine whether a kNN classifier employing BRIEF descriptors as an image preprocessing step can leverage spatial bias to improve classification performance.

We also implemented two kNN distance metrics based on Euclidean image distance to serve as benchmarks for performance. The first is a standard L2 Euclidean distance metric. The second is based on finding and exploiting convolutional kernels, and is intended to demonstrate the impact of a strong spatial bias.

We found that, out of the six variations of the BRIEF descriptor we evaluated, two of them exhibit spatial bias. In one case, the spatial bias was positive; in the other case, it was negative, a surprising and interesting result. The convolutional variation of the Euclidean distance we employed as a benchmark also exhibited positive spatial bias. These results demonstrate that spatial bias is not, by itself, a guarantee of performance improvement; certain types of spatial bias actively hinder good performance.

## Spatial Bias

The importance of spatial relationships among the pixels within an image can be readily perceived by imagining that the pixels of a digital image have been randomly permuted. This process creates an image that is unrecognizable to a human, thus indicating that these spatial relationships are essential for human image recognition. It must then be recognized as supremely ironic that many machine learning algorithms for image classification typically ignore these spatial relationships.

A rigorous formulation of this intuition is given by (Mitchell and Sheppard 2019), who employ image permutation to demonstrate whether a machine learning algorithm exhibits a spatial bias. In their study, algorithms exhibiting a spatial bias demonstrate lower classification error rates than algorithms lacking a spatial bias. This result is intuitive given the clear utility of spatial relationships in human image recognition. Their results with introducing a positive spatial bias to a random forest image classifier additionally suggest the utility of discovering new ways of incorporating spatial bias into many different machine learning algorithms applied to image classification tasks.

## Distance Metrics

### Euclidean Metrics

Our experimental baseline is the L2 Euclidean distance metric, which we calculate as the sum of squared differences of corresponding pixel intensity values. As a straightforward pairwise metric, it exhibits no spatial bias.

| | | | | | |
|---|---|---|---|---|---|
| -1 | 0 | 1 | -1 | -2 | -1 |
| -2 | 0 | 2 | 0 | 0 | 0 |
| -1 | 0 | 1 | 1 | 2 | 1 |

Figure 1: Horizontal and Vertical Canny Edge Detectors

We also employ a convolutional variation of this metric that we developed, inspired by the convolutional layers of a convolutional neural network. (LeCun et al. 1998) In image processing, a convolutional kernel is a small mask that when superimposed on the neighborhood of a pixel generates a strong response if the pixel neighborhood matches the feature encoded by the kernel, and a weak response otherwise.

A well-known application of convolutional kernels is edge detection. By using two different kernels, one representing horizontal change and the other representing vertical change, multiplying the kernel values by the corresponding pixel intensity values, and adding up the resulting values, we can represent the ”edginess” of a pixel based on the magnitude of the response. Two such kernels are presented in Figure 1; these are the Canny edge detectors.

A weakness of this approach is that humans must hand-design convolutional kernels that are then expected to properly match the characteristics of the images that they process. A key contribution of the convolutional neural network is to employ an unsupervised learning algorithm to automatically extract from the training images a set of convolutional kernels that represent features actually present in that specific set of images. The response of each pixel neighborhood to the extracted kernels can then be measured by the L2 Euclidean distance between the neighborhood and each extracted kernel. In fact, each pixel neighborhood can be filtered through each kernel to produce an output image corresponding to that kernel. In the output image, the intensity of each pixel corresponds to the distance between the pixel's neighborhood and the kernel.

By projecting an image through $n$ convolutional kernels, we generate $n$ output images. Because these images are derived from the relationships among a pixel and its neighboring pixels, we hypothesized that applying the L2 Euclidean distance between lists of output images would have a spatial bias.

Our implementation of the convolutional concept follows the algorithm below, which produces a list of one output image per convolutional kernel.

- Use k-means++ (Arthur and Vassilvitskii 2007) to create eight clusters from all of the 3x3 subimages of all of the training images. These clusters serve as our convolutional kernels.
- For each image
    - For each convolutional kernel (i.e., the k-means clusters)
        - For each pixel from the image (with a stride of 2)
            - Find the Euclidean distance between the 3x3 subimage surrounding the pixel and the current convolutional kernel.
            - Scale the distance down to the range of possible pixel values.
            - Set the corresponding pixel in the output image for that kernel to the scaled distance value.

For each input image, this produces eight output images, each of which has one-quarter the pixels of the original input image. The distance metric is the sum of the Euclidean distances between the eight output images of each original image being compared.

## BRIEF Metrics

A BRIEF descriptor is a bit vector generated from an input image using a list of test pairs of (x, y) coordinates. If the pixel intensity of the first pair is less than that of the second pair, the BRIEF descriptor value for that pair is set to one; otherwise, it is set to zero. The bit vector representation is highly appealing. The Hamming distance between two images represented as BRIEF descriptors is calculated by finding the exclusive-or of the two bit vectors and then counting the number of bits set to one.

For all of our experiments, we employed 6,272 pairs of pixels (28x28 image, eight pairs per pixel) to calculate each BRIEF descriptor. The pairs were generated in the following ways, the first two of which (denoted “Classical”) were introduced in (Calonder et al. 2010), the last four of which were devised for this paper:

1. **Uniform Classical BRIEF**: Select each x and y coordinate with uniform probability.
2. **Gaussian Classical BRIEF**: Select each x and y coordinate from a Gaussian distribution. We set the mean at the center of the image and the standard deviation as one-sixth of the size of the side of the image. This introduces a bias in favor of points closer to the center of the image.
3. **3x3 Neighbor BRIEF**: For each pixel in the image, create a pair between that pixel and each of the eight pixels neighboring that pixel.
4. **Uniform Neighbor BRIEF**: For each pixel in the image, select eight other pixels with uniform probability.
5. **Gaussian Neighbor BRIEF ($\frac{1}{3}$)**: For each pixel in the image, select eight other pixels from a Gaussian distribution. The mean of the distribution is the location of the pixel. The standard deviation is one-third the size of the side of the image.
6. **Gaussian Neighbor BRIEF ($\frac{1}{7}$)**: Same as above, except with standard deviation at one-seventh the size of the side of the image.

Aside from 3x3 Neighbor BRIEF, all of the lists of test pairs were randomly generated. The lists were generated only once; every training and testing image was filtered using the exact same list of test pairs for each algorithm. Because each pair describes a relationship between the intensity values of two distinct pixels at different locations in the image, we hypothesized that BRIEF distances in general would exhibit spatial bias.

| Error Rates (%) | | |
|---|---|---|
| Distance Metric | Original | Permuted |
| Euclidean | 3.12 | 3.08 |
| Convolutional Euclidean | 2.72 | 6.09 |
| Uniform Classical BRIEF | 3.47 | 3.54 |
| Gaussian Classical BRIEF | 3.98 | 5.54 |
| 3x3 Neighbor BRIEF | 5.42 | 3.39 |
| Uniform Neighbor BRIEF | 3.44 | 3.47 |
| Gaussian Neighbor BRIEF ($\frac{1}{3}$) | 3.23 | 3.44 |
| Gaussian Neighbor BRIEF ($\frac{1}{7}$) | 3.47 | 3.41 |

Table 1: MNIST Error Rates (%)

| Execution Times (s) | | |
|---|---|---|
| Distance Metric | Conversion | Evaluation |
| Euclidean | 0 | 914 |
| Convolutional Euclidean | 3358.00 | 15414 |
| Uniform Classical BRIEF | 3.18 | 651 |
| Gaussian Classical BRIEF | 4.17 | 621 |
| 3x3 Neighbor BRIEF | 3.76 | 655 |
| Uniform Neighbor BRIEF | 3.24 | 622 |
| Gaussian Neighbor BRIEF ($\frac{1}{3}$) | 3.25 | 637 |
| Gaussian Neighbor BRIEF ($\frac{1}{7}$) | 3.40 | 640 |

Table 2: Execution Times (s)

## Experimental Evaluation

The full source code for these experiments is available for download and experimentation: `https://github.com/gjf2a/flairs33`.

For all experiments, we employed $k$-nearest-neighbor with $k = 7$ on the MNIST database of handwritten digits (LeCun, Cortes, and Burges 1998). Image permutation was implemented by shuffling an array containing all of the pixel indices and then reordering the pixels according to the shuffled index ordering.

From the data in Figure 1, we can see that Euclidean, Uniform Classical BRIEF, and the Gaussian and Uniform Neighbor BRIEF variants are largely permutation-invariant, and hence lack spatial bias. Convolutional Euclidean and Gaussian Classical BRIEF perform considerably worse on the permuted images, thus showing a positive spatial bias. 3x3 Neighbor BRIEF performs considerably better on the permuted images, showing a negative spatial bias.

The Convolutional Euclidean has both a fairly drastic error rate difference in comparison to the permuted image, and is also the only metric we evaluated with a lower error rate than the simple Euclidean. This demonstrates that the convolutional approach successfully encodes spatial information in the kernel projections, as we would have expected given the success of this approach with convolutional neural networks.

Gaussian Classical BRIEF performs a Gaussian sampling that prefers to select pixels near the middle of the image. In the handwriting samples in the MNIST data set, most of the distinctive information is located closer to the middle of the image; the pixels towards the boundaries are almost all background pixels. By permuting the images, many pixels with distinctive information are exiled to the boundary areas of the image which Gaussian Classical BRIEF ignores. The comparison with the very similar Uniform Classical BRIEF metric is illuminating. Since Uniform Classical BRIEF selects its pixel pairs uniformly, permutation has demonstrably minimal impact on its performance.

The negative spatial bias of 3x3 Neighbor BRIEF is attributable to the fact that selecting pixel pairs of exclusively consecutive pixels prevents pairings that reflect associations between spatially disparate pixels. From this, we learn that associations between spatially disparate pixels are essential for good BRIEF performance. Permuting the image effectively transforms the metric into Uniform Classical BRIEF, with a highly similar error rate. This provides an effective solution to the problem of negative spatial bias, although it does not then further introduce any noticeable positive spatial bias.

The Gaussian and Uniform Neighbor BRIEF variants also lack spatial bias. This was more surprising in the case of the Gaussian Neighbor BRIEF variants, where the intention was to offset some of the problems with 3x3 Neighbor BRIEF by encouraging pixels to be nearby but allowing more distant pixels to play a role as well. These results demonstrate that the pixel pairs simply need to be able to represent various parts of the image, and that if they do so there will be success. The precise locations of the pairs are not as important as the idea that there is a broad sampling from among the possibilities. 3x3 Neighbor BRIEF represents an unhelpful constraint on those possibilities.

Figure 2 shows the time necessary to run these experiments on a Dell Latitude 5289 Laptop with an Intel Core i5-7300 CPU and 8 GB of RAM. The conversion times reflect the time necessary to generate the target representation for all of the training and testing images, whether it be a BRIEF descriptor or a list of convolutional output images. The evaluation times show the time necessary to compare each of the 10,000 MNIST testing examples against each of the 60,000 MNIST training examples in order to perform a kNN classification. All of the BRIEF variants run considerably faster than the Euclidean distance metric. This is attributable to optimization opportunities made available by the bit vector representation of the BRIEF descriptors.

It is not surprising that the Convolutional Euclidean is by far the slowest. Due to its strong performance, we would like to test it with two iterations of the convolutional kernels (yielding 64 7x7 images), but we have not yet had the computational resources available to complete this test.

## Conclusions and Future Work

BRIEF descriptors are an appealing source of distance metrics due to their high potential for rapid calculation. In this work, we showed that BRIEF-described metrics can perform comparitively to a traditional L2 Euclidean metric. This is all the more interesting because of the large amount of intensity information they discard. Furthermore, although the spatial relationships between pixels are crucial to the success of these methods, in general they do not exhibit spatial bias. In the cases where they do exhibit spatial bias, that bias is a consequence of an uneven distribution for the selection of the test pairs.

One important lesson from this work is that spatial bias is not uniformly beneficial. A distance metric can become overspecialized as a result of its spatial bias, limiting its performance in comparison to approaches that leverage larger samples of image pixels. The importance of this lesson can be easily overstated, however, especially in light of the strong performance of the Convolutional Euclidean metric. That metric demonstrates the enormous potential that a properly configured spatial bias can unlock.

As this work represents only a preliminary investigation, there are multiple avenues we plan to explore from here. First of all, we plan to investigate the utility of a second iteration of the convolutional process for the Convolutional Euclidean metric. We are concerned about performance issues but we have some concrete plans in place to ameliorate them, including the potential exploitation of the GPU, as has long been undertaken with Convolutional Neural Networks.

Second, the BRIEF descriptors were motivated in part by the need to avoid problems with lighting variations that are common in robotic sensing environments. Using inequalities to represent only comparative intensity differences goes a long way to neutralize this. Consequently, we plan to repeat these experiments using image data sets derived from robots in realistic environments. This could yield quite different results in comparison with the highly engineered MNIST data set. In particular, it will give the BRIEF descriptors a chance to shine in the type of environment for which they were invented.

Third, we plan to combine these approaches to see what kind of results we might yield. We would like to use the BRIEF descriptors to generate binarized versions of the input images to which the convolutional approach could be applied. Our hope is that the high performance potential of the bit vectors when properly optimized could enable the convolutional approach to become practical on the small embedded devices frequently used for controlling robots. Considerable additional thought and experimentation will be necessary to achieve this, but we believe the present work lays a solid foundation for doing so.

Fourth, kNN is widely seen as impractical to deploy due to its huge memory requirements. However, a practical classifier can be built from an unsupervised clustering algorithm which places a constant bound on the number of examples to be stored. One could, for example, employ k-means++ to determine an ideal set of examples from an arbitrarily large data set to employ as kNN inputs. The Kohonen Self-Organizing Map (Kohonen and Honkela 2007) can likewise be suitable for this purpose. We plan to evaluate this approach as well in our ongoing work.

Other kNN distance metrics also use spatial information, such as the Tangent Distance (LeCun et al. 1998) and shape context matching (Belongie, Malik, and Puzicha 2002). Our future work will explore the relationship between the concepts described in this paper and those metrics.

The full source code for these experiments is available for download and experimentation: `https://github.com/gjf2a/flairs33`.

## References

Arthur, D., and Vassilvitskii, S. 2007. K-means++: The advantages of careful seeding. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, 1027–1035.

Belongie, S.; Malik, J.; and Puzicha, J. 2002. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Calonder, M.; Lepetit, V.; Strecha, C.; and Fua, P. 2010. Brief: Binary robust independent elementary features. In *Proceedings of the 11th European Conference on Computer Vision*, 778–792.

Kohonen, T., and Honkela, T. 2007. Kohonen network. *Scholarpedia* 2(1):1568. revision #122029.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278–2324.

LeCun, Y.; Cortes, C.; and Burges, C. J. 1998. `http://yann.lecun.com/exdb/mnist/` Retrieved November 18, 2019.

Mitchell, B. R., and Sheppard, J. W. 2019. Spatially biased random forests. In *Proceedings of the 32nd International Florida Artificial Intelligence Research Society Conference*, 20–25.

Peterson, L. E. 2009. K-nearest neighbor. *Scholarpedia* 4(2):1883. revision #137311.