# Recognising Image Shapes from Image Parts, not Neural Parts

KIERAN GREER,
Distributed Computing Systems, Belfast, UK.

*Abstract:* This paper describes an image processing method that makes use of image parts instead of neural parts. Neural networks excel at image or pattern recognition and they do this by constructing complex networks of weighted values that can cover the complexity of the pattern data. These features however are integrated holistically into the network, which means that they can be difficult to use in an individual sense. A different method might scan individual images and use a more local method to try to recognise the features in it. This paper suggests such a method and it is conjectured that this method is more 'intelligent' than a traditional neural network. The image parts that it creates not only have more meaning, but they can also be put into a positional context and allow for an explainable result. Tests show that it can be quite accurate, on some handwritten digit datasets, but not as accurate as a neural network. The fact that it offers an explainable interface however, could make it interesting.

*Key-Words:* image classifier, image part, quick learning, positional context, explainable.

Received: May 27, 2022. Revised: July 23, 2023. Accepted: August 25, 2023. Published: October 2, 2023.

## 1 Introduction

This paper describes an image processing method that makes use of image parts instead of neural parts. It is conjectured that this method is more 'intelligent' than a traditional neural network, where the image parts that it creates not only have more meaning, but they can also be put into a positional context and allow for an explainable result. Neural networks excel at image or pattern recognition and they do this by constructing networks of weighted values that can cover the complexity of the pattern data. These networks recognise similarities in the data and resolve that into features which are shared between the patterns. These features however are integrated into the network, which means that changing a feature can have unexpected consequences and they can be difficult to use in an individual sense. While tests indicate that a holistic view is still the best, it is the incomprehensible nature of the network nodes that is the key factor.

A different method might scan individual images and use a more local method to try to recognise the features in it. This paper suggests such a method, where a trick during the scan process can not only recognise separate image parts, as features, but it can also produce an overlap between the parts. This is very helpful and it means that the image parts can be placed into a positional context with each other. Then when comparing with a new image, it can be similarly parsed, when the image parts also need to be in the same relative position, to be compared with each other.

The process is intended to recognise image shapes, more than internal colours or textures, but this is still a difficult and challenging task. The tests of section 4 have been carried out on handwritten digit characters, where it is noted in [6] that handwritten character classification is fundamental for postal sorting, bank check recognition, automatic letter recognition, industrial automation, human-computer interaction, and historical archive documents. Initial tests suggest that this new method is reasonably accurate but may require improvements to compete with state-of-the-art. It does however, fit well with the author's own cognitive model [8], as part of a symbolic system.

The rest of this paper is organised as follows: section 2 gives some related work, while section 3 describes the new classifier in more detail. Section 4 describes some implementation details and test results, while section 5 gives some conclusions to the work.

## 2 Related Work

Deep Learning [9][10][12] has managed to almost master image recognition, but Decision Trees [4] are not far behind. At the heart of Deep Learning and the original Cognitron, or

Neocognitron architectures [5], is the idea of learning an image in discrete parts. Each smaller part is an easier task and cells can then be pooled into more complex cells with neighourhoods. The deep learning architecture of [9] ends up with a top two layers that form an undirected associative memory, for example. Or a convolution can exaggerate a feature through a local transformation, to convert an image into one that represents the feature more.

A different algorithm was tried in [3] to recognise the letters dataset used later in section 4. They used a bag-of-visual-words, where objects are represented as histograms of feature counts. While shape is one feature, missing from this might be relative position. In fact, recognising the image shape is quite an old idea, where lots of evaluation formulae are available. It probably pre-dates using neural networks, where one summary could be [16]. The paper [6] describes some other shallow architectures that include convolutions. They suggest a new Fukunaga–Koontz network that would process images more orthogonally and locally, but with the more advanced neural network architecture. However, the paper does recognise the goals of this paper when producing their new network structure.

The image classifier has derived from earlier work by the author, including the papers [7][8]. The paper [8] gives a first version for the algorithm, using only cell relations. Treating each pixel as a cell requires it to have a weighted association with the other pixels, which in that paper spanned the whole image. For example, a grid cell would map to all the other cells in an image it was present with, as a type of cross-referencing, to represent the cell importance with the desired image category. There is no overlap with cells only 1 pixel in size, but mapping the cells can give the region some definition that can make it both distinct and allow for overlap. Using these local mappings therefore, can also exaggerate associations, depending on how the weight update is performed. A second paper [7] then showed that local mappings can produce a reasonably useful auto-associative classifier. It also showed that the local calculation can replace the fully-connected weight values and produced state-of-the-art results on one dataset. That idea has then been used in this paper to produce the image parts, as described in section 3.1.

There is some evidence that the scanning process of this method may mimic human eye movements. There are different types of eye movement [2], including smooth tracking movements or saccadic irregular movements, to fixate on and recognise features. These more irregular movements are what the new algorithm would make use of. It is interesting that the paper also writes about neural binding, as part of feature integration, which has also been studied as part of the author's cognitive model.

The paper [11] suggests using attentional models instead of deep neural networks. It states that the computational expense of neural networks scales with the dimensionality of input images and can become prohibitive. Attentional models recast computer vision as a sequential decision-making problem, allowing an agent to deploy a sensor (i.e., an attentional window) to image data across multiple time-steps and the approach bears a resemblance to perceptual psychology. The paper's results demonstrate that carefully chosen models of visual attention can increase not only the efficiency, but also the accuracy of scene classification.

Another paper that might have biological relevance is [13], which suggests that the hierarchical organization of the human visual system is critical to its accuracy. This is a functional hierarchy rather than a tree shape, however. While neural networks can learn this, they require orders of magnitude more examples than a human, who can accurately learn new visual concepts from sparse data, sometimes just a single example. Inherent in this then is the idea of orthogonality, but they do still use deep learning as part of the architecture, to build the hierarchy of prior knowledge and exemplars.

# 3 The Image Recognition Classifier

This section describes the image recognition classifier, which splits images into parts and then associates output categories with the parts, to define the clusters. Then to create the classifier and help to economise, the image part descriptions can be clustered into exemplar sets for each category type, when an exhaustive search over these exemplars can classify previously unseen images with reasonable accuracy. The self-organising process to generate exemplars however, only reduced the total number of images by a small amount, and so the testing more often compared examples with individual parsed images instead.

## 3.1 Image Parts

The new algorithm is based on the fact that scanning over an image will automatically separate its shape into discrete parts, but it depends on the angle at which the image is scanned. If the scan is done in a vertical or a horizontal direction, then the

whole image is returned and so angular scans (North-West, South-East, etc.) are required. It might be along the lines of irregular eye movements, for example. The current method is only useful for describing distinct features in the outer image shape and would not be useful for recognising internal patterns or colour, for example. But one aspect of it is that the parts can recognise regions where lines join with each other. This type of recognition might require real intelligence in a different system, but the scanning trick is able to realise this secondary feature for itself. The algorithm is a bit complicated to describe in detail, so a very general overview is as follows:

1. Scan the images in the different directions (NW, NE, SW, etc).
2. Generate combined lists of node co-ordinates and distances, from the start to the end of horizontal or vertical lines in those directions and create an image part from them.
3. Re-order the image parts on decreasing size.
4. Some parts are contained in other ones, so remove any contained parts.
5. What is left represents the image in parts and the relative positions of the parts can also be stored.

Figure 1 is a nice example that shows the image parts generated for a number '4'. One part is not included, due to its insignificant size.

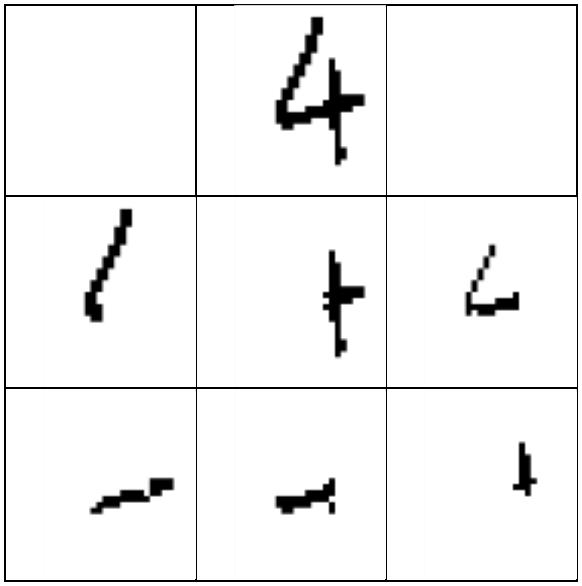

Figure 1. Image parts (rows 2 and 3) generated for a number 4 (row 1).

The scanning process makes use of the ideas of a continuous sequence and also convolutions [6][10], or producing an aggregated score from a region. The idea of cell associations was central to the first algorithm of [8] and subsequent work, and it is really only a count of what other cells are present when the cell in question is present. For recognising features, the scan counts the number of continuous cells before an empty cell is encountered, in the indicated direction. This helps to recognise the lines in the image, where the distances are then aggregated into cohesive sets. But at the moment, this is only for binary images with a 1 or 0 value in each cell. After a scan in a particular direction, cells with similar scores can be grouped together as an image part, and in fact, scans in different directions can also be cumulated together. In Figure 1, the image parts in rows 2 and 3 can be of different sizes, where they also overlap and the overlap can include joining regions, such as where the main horizontal and vertical lines join.

### 3.2 Relative Positioning

The image parts for a whole image can therefore be placed in order of their size and then what they link to. This can be done for each image individually, making it orthogonal and it is a very explainable process. Because the parts are easily recognised in the original image, their relative position can also be determined. For the current implementation, the full image is divided into 16 regions, where a 32x32 pixel image would be

divided into 16 8x8 regions, for example. Each image part that links to another part has a positional array that stores a value for the 'North-South-West-East' directions. If the image part is positioned at the center of the image it links with, then all the values are 0. It can then either be 1 or 2 steps away in any of the 4 directions and this can be easily determined.

This helps to put the image part into context with the larger part and when comparing images, the parts should have the same relation with their larger counterparts to be considered. Note that this does not require an exact positional match, but has some leeway as to where exactly the parts are placed. It would mean, for example, that an image with a line at the bottom would not be confused with an image with a line at the top, but two lines at the top do not have to be exactly in the same place. It also means that, for the classification process, the image parts can be cropped before being stored, because their relative positions are translated over to the positional array.

# 4 Implementation and Testing

A computer program has been written in the Java programming language. It is able to convert binary images into ascii 1-0 representations. These were then read into train and test datasets for each category. After reading the image data, each image can be defined by a set of image parts. It would be possible to use the images like that, or it is possible to try to create exemplars from them. This may reduce the number of images to consider, when searching for a category. Two different methods were therefore tried – one that used exemplars and a wholly distributed method, as described next. Both methods were deterministic, meaning that they always returned the same result.

## 4.1 Using Exemplars

Clustering the images into exemplars can be a self-organising process as follows: A distance can be measured for the closest images between categories. Then when combining images inside of a category, the distance between them must be less than the minimum distance to any image in any other category. This however, reduced the total number of images by a small amount only, replacing some by an aggregated result that was an exemplar.

The train images were learned very quickly and most of the time was taken when trying to classify the test images, which was an exhaustive comparison with all the exemplars/images. For these small image sets however, processing a test image required only a few seconds. Then a count of the actual versus the closest category match for each test image was done, resulting in a percentage accuracy score. A similarity score to an exemplar was therefore performed as follows:

1. Measure the similarity of the pixels in two shape parts.
2. If they are in the same relative position, then add only the difference in the parts to the score. If they are in different positions, then add the total pixel counts to the score.
3. Only use a shape part once and always try to match parts on position.
4. The lowest score indicates the best match.

## 4.2 Distributed Mapping

A second and probably simpler method mapped directly from a part to a list of output categories. A train image would be parsed and the parts matched with a database of all parts. The relevant part would be retrieved and the output category added to it. A test image would then be parsed into its' parts and these would be matched directly to the database of all parts. This would return lists of associated output categories and the category with the largest count overall would be selected. The results for this method however, were not as good as for exemplars.

## 4.3 Hand-Written Numbers Datasets

A first test used the Chars74K set of hand-written numbers [15], but only the numbers 1 to 9. There were approximately 55 examples of each number and the binary image was converted into a 32x32 black and white ascii image first. The examples were then divided into a train set of 40 images and a test set of 15 images. After exemplars were learned for the 9 train categories, each of the 15 images in the 9 test sets were compared and matched with their closest category. This was an exhaustive search process, where one version classified the test images to an average accuracy of 80%. As a comparison, an earlier image recognition attempt [8] only produced a 46% accuracy over the same dataset. The Deep Learning methods however, are able to recognise the number sets, essentially to 100% accuracy ([9] and more recent). A second test used the Semeion Handwritten Digits Dataset [1][14], with a split of 120 images in the train set and 40 images in the test set. The dataset contains 1593 handwritten digits from 0 to 9, converted into 16x16 black and white ascii images. One version classified the test images to an average accuracy of 75%. The original paper [1] quoted a success score of about 93%, where mid-90% is also quoted in other papers. Note however, that the auto-

associative classifier in [7] scored these datasets at 96% and 99.8% accuracy respectively. It looks like some accuracy may be lost when moving from pixel-related associations to larger shapes. The system in [3], for example, only scored about 55% accuracy for the Chars74K dataset, but that was for all of the symbols and not just the numbers.

# 5 Conclusions

This paper describes a new image-processing algorithm that is very human-like. It processes and stores images individually, but these can then be clustered into exemplars. The process uses something resembling an eye-scan which moves in angular directions. It is conjectured that the image parts are more 'intelligent,' because they are more explainable. The process can even include information about the relative positions of each part.

The method is shown to be very quick for small image sets, but it requires an exhaustive search over all saved exemplars, which might require some sort of heuristic search, if the database was to grow very large. However, if it cannot be as accurate as cell-based or neural networks, for example, then part of the human learning process must be missing, or maybe some refinement is still required. A second distributed test did not fare quite as well as using exemplars and so the conclusion here is that the holistic view is still more important, but that finer details are also required.

The advantage of the method is the fact that it is explainable. The image parts can be used at a symbolic level, for example, where they could be integrated with other types of data. This might be a false goal, if an AI system ultimately needs to process at a neural level, and the parts are not always the most meaningful. But it would at least allow the symbolic processes to be studied, across data types. The system is deterministic and always returns the same result, which might be an interesting property of a symbolic system over a neural one, in that it can add some stability.

*References*

[1] Buscema, M. (1998). MetaNet: The Theory of Independent Judges, *Substance Use & Misuse*, 33(2), pp. 439 - 461.

[2] Chen, K., Choi, H.J., and Bren, D.D. (2008). *Visual Attention and Eye Movements*.

[3] de Campos, T.E., Babu, B.R. and Varma, M. (2009). Character recognition in natural images, In *Proceedings of the International Conference on Computer Vision Theory and Applications* (VISAPP), Lisbon, Portugal.

[4] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K. and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248-255.

[5] Fukishima, K. (1988). A Neural Network for Visual Pattern Recognition. *IEEE Computer*, 21(3), 65 - 75.

[6] Gatto, B.B., dos Santos, E.M., Fukui, K., Junior, W.S.S. and dos Santos, K.V. (2020). Fukunaga–Koontz Convolutional Network with Applications on Character Classification, *Neural Processing Letters*, 52, pp. 443 - 465. https://doi.org/10.1007/s11063-020-10244-5.

[7] Greer, K. (2022). Image Recognition using Region Creep, *10th International Conference on Advanced Technologies* (ICAT'22), pp. 43 – 46, November 25-27, Van, Turkey. Virtual Conference.

[8] Greer, K. (2018). New Ideas for Brain Modelling 4, *BRAIN. Broad Research in Artificial Intelligence and Neuroscience*, 9(2), pp. 155-167. ISSN 2067-3957.

[9] Hinton, G.E., Osindero, S. and Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets, *Neural computation*, 18(7), pp. 1527 - 1554.

[10] Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097-1105.

[11] Kuefler, A. (2016). Attentional Scene Classification with Human Eye Movements, *http://cs231n.stanford.edu/reports/2016/pdfs/000_Report.pdf*. (last accessed 18/7/23).

[12] LeCun, Y. (2015). What's Wrong with Deep Learning? In *IEEE Conference on Computer Vision and Pattern Recognition*.

[13] Rule, J.S. and Riesenhuber, M. (2021). Leveraging Prior Concept Learning Improves Generalization From Few Examples in *Computational Models of Human Object Recognition, Frontiers in Computational Neuroscience*, 14, Article 586671, doi: 10.3389/fncom.2020.586671.

[14] *Semeion Research Center of Sciences of Communication*, via Sersale 117, 00128 Rome, Italy, and Tattile Via Gaetano Donizetti, 1-3-5,25030 Mairano (Brescia), Italy.

[15] The Chars74K dataset, *http://www.ee.surrey.ac.uk/CVSSP/demos/chars74k/*. (last accessed 18/7/23).

[16] Yang, M., Kidiyo, K. and Joseph. R. (2008). A Survey of Shape Feature Extraction Techniques. *Pattern Recognition*, 15(7), pp. 43-90.

**Acknowledgement**

The author would like to thank the reviewers for their helpful comments.

**Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)**

The author contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

**Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself**

No funding was received for conducting this study.

**Conflict of Interest**

The author has no conflict of interest to declare that is relevant to the content of this article.