Two Dimensional Intelligent Character Recognition

Ryan Pattison, Douglas Anderson, and Oliver Cook[a,b,c]

[a] *ryan.m.pattison@gmail.com*
[b] *dander01@uoguelph.ca*
[c] *cooko@uoguelph.ca*

Abstract

Intelligent Character Recognition (ICR) has become a classic machine learning problem with many successful algorithms for classifying both printed and hand written text. In this paper, we create a classifier which classifies texture based symbols of terrains on maps in two dimensions. Three classification methods are implemented including a feature-based approach, a gold standard image comparison method, and an approach based on symbol proximity. The results show that the proximity based approach, presented here, can predict with high confidence, 94%, the identity of an unknown symbol, using only the surrounding symbols.

Keywords: Machine Learning, Data Mining, Computer Vision, Intelligent Character Recognition

1. Introduction

The problem of Intelligent Character Recognition (ICR) in Artificial Intelligence and Machine Learning has received much attention in past years. Generally this field has focused on the classification of *handwritten* or *typewritten* letters and numbers. The novelty of this project has been in the development of an algorithm which recognizes characters, or more generally symbols, arranged to form words in *two* dimensions as opposed to recognizing linear text. The practical application proposed is the development of an algorithm for recognizing hand drawn cartography. To this end, we have defined a small set of symbols to represent various terrains: grass, mountains, bodies of water, and more. Each symbol will be drawn into a single cell on a standard sheet of grid paper and scanned into a computer. Using the scanned images, several algorithms will learn to recognize the symbols cell-by-cell as well as in the context of the surrounding cells. To demonstrate the utility of such an algorithm, we propose an application which reads a hand drawn map and produces a computer generated image of the same map. To do this, we have used the software tool WEKA[1] to assist us in the creation of various classifiers.

Figure 1: Pallet Town map from Pokémon FireRed [2]

Table 1: Symbol names and the hand drawn forms

Water		Grass	
Rock		Tree	
Dirt		Sand	

2. Process

2.1. Overview

The maps used for classification in this project are hand drawn on standard 4 to 1" grid paper using a custom set of symbols defined for this purpose. These maps vary in cell dimensions and distribution of symbols but are all derived from the popular video game Pokémon. This was done to ensure that proximity information and terrain types were standard throughout. In total, 10 different maps were used with a total of 13488 cells each containing one symbol. An example map from the game is shown in Figure 1. The custom symbol set consists of 7 symbols created to represent the terrain types needed to build the maps in Pokémon. These symbols are: water, dirt, grass, building, rock, mountain and sand and can be seen in 2.1 . Each symbol was made to be visually distinct from each of the other symbols to help with classification. By choosing a specific source such as this we ensure that our dataset is sampled from a well defined distribution.

2.2. Preprocessing

Before analysis, the data must be preprocessed in order to standardize image properties and reduce the risk of errors in classification. A collection of scripts are run on each of the

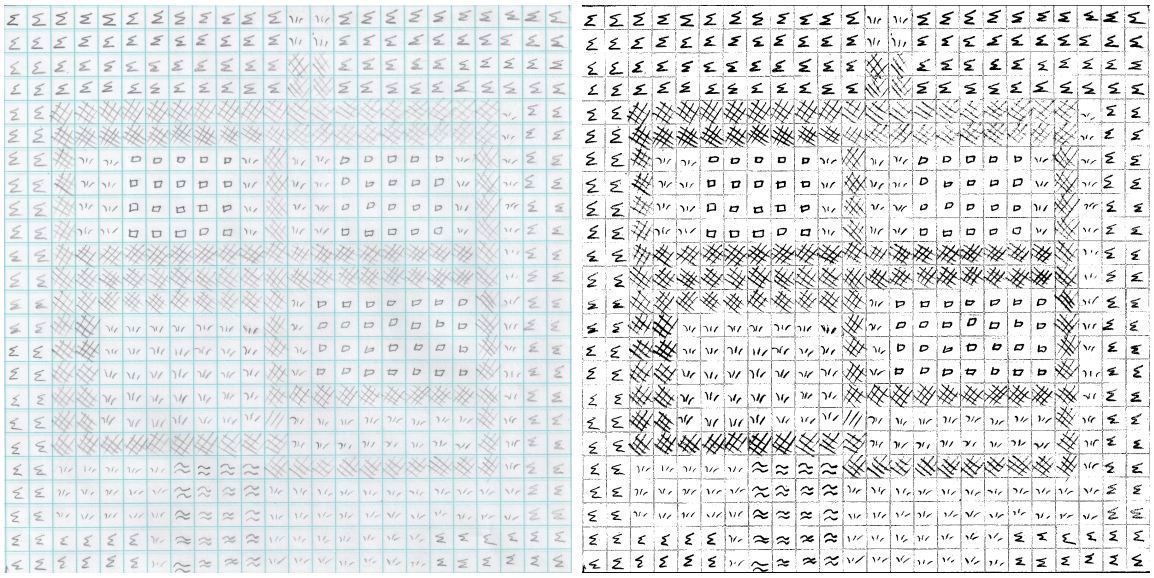

Figure 2: Image before and after preprocessing. Note the increased contrast and faint grid lines

maps before classification to rotate, scale, and remove the blue grid lines using colour filter masks. Colour information is removed from the images by converting them to black and white. Contrast is increased making the symbols appear more pronounced for classification. Each cell is scaled to a standard 75 by 75 pixels and seperated from the map so that each cell contains a single symbol. The results of preprocessing are shown in Figure 2.2

2.3. Feature-Based Approach

Feature based classification, as the name suggests, involves the classification of unknown cells through the use of geometric information regarding the shape and distribution of points from its contained physical symbol. To perform this task a collection of convolution filters were applied to each of the preprocessed cells. This was followed by statistical analysis on the distribution of black pixels in each of the of the x and y dimensions. These convolution filters are used to transform an image into another image which may make certain attributes, such as edges, more prominent. For example, the convolution filter, F_{blur} is a mean blur. The convolution filter transforms the image by replacing each pixel with a new filtered pixel. The new pixel is created by multiplying the surrounding values by those in the convolution filter and summing the result. This sum is the new value for the pixel. Each convolution filter consists of a matrix of odd size containing relevant coefficients to the feature that is being tested. In total nine filters were used, each extracting a different feature.

$$F_{blur} = \begin{bmatrix} 1/16 & 2/16 & 1/16 \\ 2/16 & 4/16 & 2/16 \\ 1/16 & 2/16 & 1/16 \end{bmatrix}$$

A filtered image is created as the result of the convolution matrix being applied to each pixel. Figure 3 shows the filter being applied to a single pixel. The filtering process repeats

$$\begin{array}{ccccc} \ddots & \vdots & \vdots & \vdots & \iddots \\ \dots & 64 & 64 & 32 & \dots \\ \dots & 32 & \boxed{32} & 128 & \dots \\ \dots & 32 & 16 & 32 & \dots \\ \iddots & \vdots & \vdots & \vdots & \ddots \end{array} \quad \begin{array}{c} 64F_{11} + 64F_{12} + 32F_{13} \\ +32F_{21} + 32F_{22} + 128F_{23} \\ +32F_{31} + 16F_{32} + 32F_{32} \end{array} \quad \begin{array}{ccc} \ddots & \vdots & \iddots \\ \dots & \boxed{48} & \dots \\ \iddots & \vdots & \ddots \end{array}$$

Figure 3: Applying the convolution filter to the centred pixel area

for each pixel until all pixels have been replaced and resulting image has a luminance matrix, which we will call A.

$$A = \begin{bmatrix} a_{00} & a_{10} & \cdots & a_{n0} \\ a_{01} & a_{11} & \cdots & a_{n1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{0m} & a_{1m} & \cdots & a_{nm} \end{bmatrix}$$

We define x and y to be the row and column sum functions.

$$x(i) = \sum_j A_{i,j} \quad y(j) = \sum_i A_{i,j}$$

Next we record the means and standard deviations: $\mu_x, \mu_y, \sigma_x, \sigma_y$ as the feature set for this filter. Then we apply another filter, either on this filtered image or, on the original image and record statistics on these as well.

2.4. Gold Comparison Approach

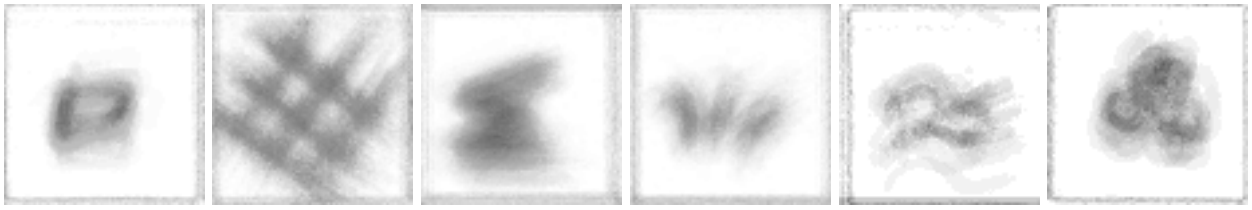

Figure 4: Mean gold images for buildings, dirt, forest, grass, water, and rocks

Gold comparison works by comparing a candidate image, A to a gold standard image G representing a possible class of A. If A is similar to G, then there is a high probability that A belongs to the class which G represents. Before comparing the images, we need to make the symbols A and G overlap so that we can more accurately compare pixel-to-pixel. To overlap the images, we define x and y to be the row and column sums of the image and calculate the mean and standard deviation of x and y, similar to the feature based approach. To get

the symbols to overlap we apply a linear transformation to each pixel using the statistics gathered in each dimension.

$$x' = (x - \mu_x^A)\,\frac{\sigma_x^G}{\sigma_x^C} + \mu_x^G \qquad y' = (y - \mu_y^A)\,\frac{\sigma_y^G}{\sigma_y^C} + \mu_y^G$$

These new transformed pixels replace those in the image A which now overlap with those in the gold image G. We then compare the overlapping image A to the gold standard image G by taking the sum of the squared difference at each pixel position. This gives us an error measure, which we record for each of the k classes.

$$\xi_k(A) = \sum_i \sum_j (A_{ij} - G_{kij})^2$$

The result is a vector, ξ with an error measure for the comparison against each class. For training purposes, we then input this vector into WEKA along with the correct answer.

To create the gold images, we collect all the samples in our data set and create a mean gold image by overlapping all the samples belonging to the same class, the results can be seen in Figure 4 on page 4.

2.5. Proximity Approach

The proximity classifier learns the patterns of symbols present in the data set, and then classifies any unknown symbols using the surrounding symbols, and its knowledge of the patterns and relationships of the symbols. Here we begin by outlining the methodology used; we want to classify the symbol x at a given position, as Figure 5 illustrates.

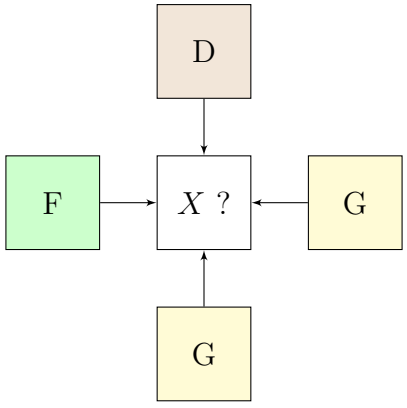

Figure 5: This example is from cell 2,19 in Pallet Town, Note: G=grass, D=dirt, F=forest

To classify x, we first calculate the probability that x belongs to each class, $P(X) \quad X_i = P(x \in C_i)$. The proximity classifier is given the class of each symbol in the cardinal directions: north, south, east, and west. Using this information, we restrict the probability calculation

	c_B	c_D	c_{EDGE}	c_F	c_G	c_W
north	0	0.1558	0.0519	0.0519	0.7403	0.0325
east	0.0779	0.0974	0	0.1493	0.6428	0
south	0	0.2403	0.0130	0.0065	0.7403	0
west	0.0519	0.2273	0	0.0519	0.6428	0.0260

Table 2: Probability of the neighbours for a grass cell in Pallet Town

to consider only the case where a symbol has the given classes neighbouring it. Such a restriction is the conditional probability:

$$P(X) = P(X|\, N{=}c_1,\, E{=}c_2,\, S{=}c_3,\, W{=}c_4)$$

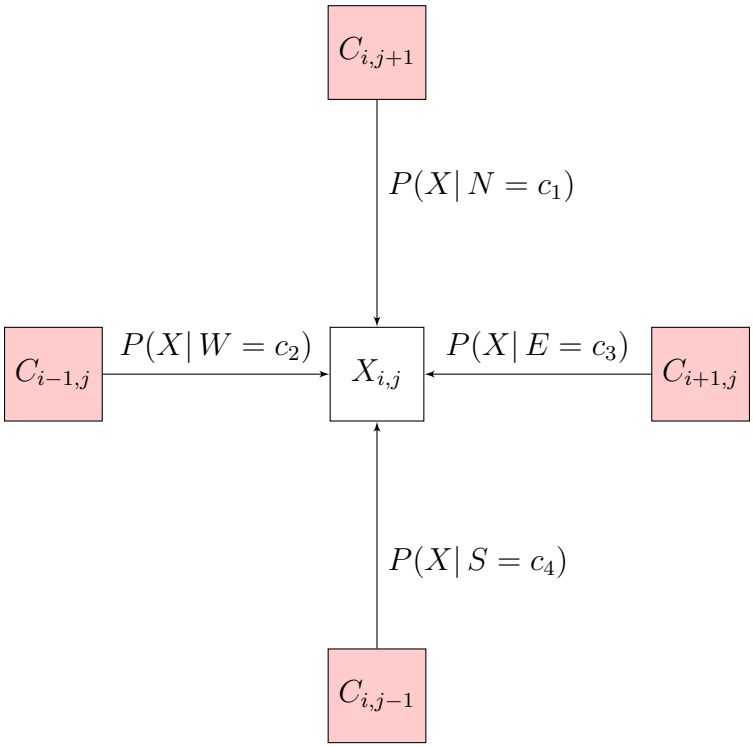

Figure 6: The conditional probability of the classes for $X_{i,j}$, given the neighbouring cells

In calculating probabilities, we can only approximate them using relative frequencies collected from our sample data. For a set of k classes, there are k^5 possible combinations of cells, 4 neighbouring cells and the centre, and too many of these combinations occur rarely in our dataset to make confident approximations of the true probabilities. To resolve this problem, we assume that the conditional probabilities are independent in each direction. Under this assumption, the calculation becomes much more reasonable for our small data

set, since there are only $4k$ pieces of data to record for each symbol. Our resulting calculation is given in Equation 1.

$$P(X) \approx P(X|\, N{=}c_1)P(X|\, S{=}c_2)P(X|\, E{=}c_3)P(X|\, W{=}c_4) \tag{1}$$

From the definition of conditional probability we have that for some class c in the direction D:

$$P(X|\, D{=}c) = \frac{P(X \cap D{=}c)}{P(D = c)}$$

To calculate each of the conditional probabilities, we compute a table from the samples recording $P(X{\cap}D{=}c)$ and $P(D{=}c)$ for each class c, and direction D, an example is given for a single class in Table 2.5. We use the table to calculate the relative conditional frequencies and, under the assumption of independence, we approximate $P(X)$ using Equation 1.

Having the probability vector $P(X)$ we create an input file for WEKA. This file includes the probability vector, the neighbouring classes, and the class of the symbol being classified for training. WEKA creates a decision tree which is used to reduce the error in the probability calculation that is introduced under the assumption of independence.

3. Results

The following results are from running WEKA's J48 decision tree algorithim on the training set and using 10-fold cross validation. In addition to using 10-fold validation, the proximity results are from a map that was not used in training.

Algorithm	Samples	Feature Set	Success	Features	Tree nodes
Feature-Based	9888	Single Filter	76.33	31	1525
Feature-Based	9888	Single Filter+Blur	77.01	31	1465
Feature-Based	9888	No Spatial	80.64	205	1131
Feature-Based	9888	Spatial	82.47	207	1005
Proximity	13200	Surrounding	93.78	10	93
Gold Comparison	225	Max Probabilities	70.40	7	N/A
Gold Comparison	225	Probabilities	83.33	7	45

Table 3: Classification results for J48 decision trees

4. Conclusions

Each of our classification methods has its strengths and weaknesses. The feature based method is able to recognize arbitrary features and is easy to extend by adding new filters. The proximity method works extremely well with the set of maps we chose because there are many common patterns in their composition. However, to detect these patterns in a data set the proximity classifier needs an accurate representation of the map. This method has the added benefit of having its output be the same as its input, which means it can improve

its output iteratively. The gold comparison method is able to classify symbols fairly well. However, this method requires gold standard images as input and the number of features in the output is dependent on the number of gold images (which makes generalization more difficult).

5. Further Work

Future expansion of this design is possible and we believe the results are encouraging. After assessing the results, we have identified various aspects of our current design that can be changed to improve the functionality and classification results. For example, we would like to use all the methods presented here in conjunction to improve the accuracy and the robustness of the system. One such system, could run the gold comparison method to get a rough approximation of the layout of the symbols and then use the proximity method to improve the confidence in the results.

Improving on the individual methods is also possible. For example the feature-based classification may see improvement from considering other transformations of the pixels when calculating statistics, such as the distance from the centre which is especially useful for symmetrical shapes. Another improvement that we would have liked to make is to limit the region of the cell which the feature-based method analyses. By only taking statistics on these smaller sections of the images we could eliminate areas of the image that are largely noise, such as the edges of the cell. The gold comparison method could use "ideal" gold standard images rather than mean gold images and we suspect that the performance of the gold method could be improved by using carefully selected images that better illustrate the intended shape of the character. Also, adding more than one "gold" symbol for each class may allow for more variations than the mean of each variation.

Given the strength of the patterns and relationships found by the proximity statistics, the extensibility of the feature-based method, and the accuracy of the gold image classifier, we believe the results show promise in a hybrid approach of the proposed methods to create an accurate, robust, and extensible classification system.

6. References

[1] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software; an update. *SIGKDD Explorations*, 11, 2009.

[2] Nintendo. Pokémon firered, 2004. http://bit.ly/1hlq9Dy.