

COSC 4P80: Assignment 2

Experimentation and Investigation of Self-Organizing Maps

Michael Young
Brock University
St. Catharines, Ontario
Email: my08tu@brocku.ca

Abstract—Now that we're in the information age, data mining is becoming more and more popular. Self-Organising Maps turns out to be a great tool in manipulating large dimensional data and projected it in a lower dimension that makes it easier to understand. By using various forms of clustering techniques, we are able to analysis large sets of data in hopes that some hidden attribute of the data might be recognized. The number of applications for a Self-Organizing map is practically endless and in this paper we will discuss the basic problems with clustering as well as see an application for clustering of images. Naturally, will this algorithm their are various parameters that can be altered to obtain various results in the final outcome. This paper will explore some of these parameters and investigate their behaviour and effect on the final results.

Key Terms - Self-Organizing map, clustering, image analysis.

I. INTRODUCTION

The main focus of this paper is to investigate the behaviour of clustering using Self-Organising Feature Maps (SOMs) and how changes in the parameters effect the changes in map clusters. There are numerous ways to cluster information, but for the purpose of the assignment the main focus will be on Self-Organizing Maps and how it can be used for image clustering. There are a variety of parameters involved in Self-Organising Map and this paper will also investigate altering some of these parameters and determining the effect it has on the performance of the Self-Organizing Map. Some of the parameters of a SOM include Neighbourhood size and shape, learning rate, exponential decay constant and map dimensions.

II. PROBLEM

The objective of this assignment is to experiment with Kohonen Self Organizing Maps in order to determine their behaviour when using different parameters, equations, and sample types.

The problem is broken down into two main parts: Colour Organization and Image Clustering.

A. Problem 1. Colour Organization

The first problem of the paper involves color organization. This can be viewed as a toy problem or a basic example to demonstrate the behaviour of clustering, and more specifically, how SOMs themselves, can be used to perform clustering. This

is done by creating a randomly generated image, each pixel has a random RGB value and feeding random samples from this image as training examples for the network. There is further information on this problem in the Data and Samples section.

B. Problem 2. Image Clustering

The second problem involves image clustering. The goal of image clustering is to provide the network with enough training examples so that while the network is training, the images are placed in similar groups based on differing color features. Images are partitioned and information regarding the color of each pixel are encoded in a vector that will be used for training.

The purpose of grouping similar images is in hope that when the network is presented with a new image it hasn't seen before, it can properly classify it as the same type as of other images in a similar clusters. In addition to being able to classify new images, the goal of the SOM is to organize More information on Colour Organization and Image Clustering problems can be found in the Data and Samples section.

III. NEURAL NETWORKS

Artificial Neural Networks is a computational model that is inspired by biological neural networks, for example, the human brain. By connecting a certain number of neurons together to make a layer, then connecting these layers together through a series of connections each with their own weight, we are able to make a crude approximation of a biological neural network.

Using this biological inspiration, it's possible to construct multiple types of networks that have different functionalities. Of the most common is Feed Forward Neural Network. A Feed Forward Neural Network is trained using a supervised training method where the outcome of the training examples are known before hand. The output of the network is then compared to the expected output and the network is modified accordingly to help converge to these expected values.

Sometimes the information of the expected output is not always known in a data set. This is where we have to switch to a form of unsupervised learning. In the next section we will talk more about unsupervised learning with neural networks.

A. Self-Organizing Maps

Self-Organizing maps(also known as Kohonen Maps, Kohonen Self-Organizing Maps or simply SOMs) are a specific type of neural network. Originally developed by Tuevo Kohonen in 1982, SOMs are based on the associative neural properties of the brain and inspired by the grouping structure of neurological connections. The way SOMs learn and are trained in specific applications can be viewed as a form of competitive learning, as well as an added co-operative feature. This represents a form of unsupervised learning, where the outcome of training examples is not known. [1]

Another property of SOMs is their ability to transform high dimensional data into a lower dimension while still preserving the data's topology. This gives us the ability to observe possibly interesting features in data that otherwise might not be clear. [2]

There are a variety of applications for SOM's, more can be found in Kohonen's Engineering Applications of the Self-Organizing Map [3]. An interesting and widely used example can be found on ai-junkie.com [4]. In this example the quality of life was measured for a variety of countries and were fed through the SOM. These factors can include country health, nutrition, educational services etc. The outcome is a map based on the different qualities of life that were measured. As shown in figure 1, countries with similar quality of life have been clustered together into their own separate groups. It's clear from this picture that the quality of life in Belgium (Orange cluster) is drastically different that of Kenya (light blue).

With this map information, it's possible to project this information differently to get a more in depth understanding of the map and how it relates to the quality of life. Referring to figure 2 each country has been coloured according to it's cluster value of the map. This picture now shows us even more information such as where exactly on the globe that the quality of life is similar or differs.

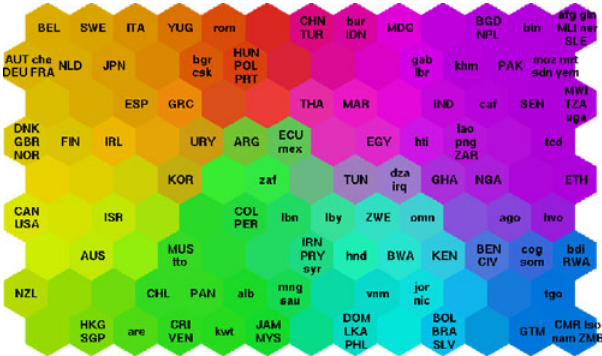


Fig. 1: Quality of Life Image Map - Countries with similar quality of life values are clustered together.

This is just one example of how Self Organising Maps can be used to represent multi-dimensional data into a lower dimensional space. Naturally, this makes understanding the problem and data interpretation much easier for humans.

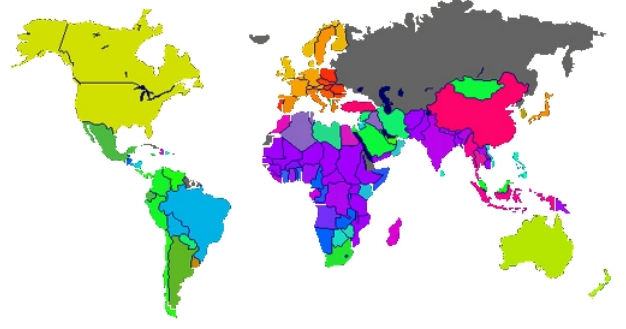


Fig. 2: Quality of Life Image Map - The colour values from 1 are displayed on the world map to get a better understanding of the proximity of countries and their corresponding quality of life values.

B. Structure

The structure of the SOM network is quite different from a Feed-Forward Neural Network. The SOM consists of an input layer connected to a 2-Dimensional lattice of neurons called the Kohonen Layer. Each neuron in the layer is connected to each neuron in the lattice. The input layer is similar to that of an input layer in a feed-forward network, where the number of neurons in the layer is equal the dimension of the sample input vector. Each neuron in the Kohonen layer contains a weight vector whose dimension is equal to the of the input vector. The topology of the network can be seen in figure 3 [5].

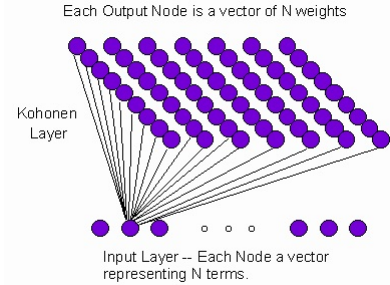


Fig. 3: Kohonen SOM Topology - An input vector is connected to a lattice of neurons each with its own weight vector.

C. Algorithm

The algorithm for SOMs is rather straight forward. Although it's difficult to express the algorithm explicitly in mathematical theorems, it's still possible to achieve an intellectual understanding of the process and outcome of a Self Organizing Map. [1]

The algorithm can be read as follows. While the clusters are still converging, grab a random sample vector from the data set and pass it through the network. The network then finds the Best Matching Unit(BMU) neuron to the sample vector and updates the weight values of this neuron as well as all the neurons in its neighbourhood by a specified value. This process is repeated until a specified number of iterations or the clusters have converged.

Algorithm 1 Self-Organizing Map

Require: Initialize the weights in the network
while time < t_{max} and convergence has not be reached **do**
 Choose random sample vector from data
 Find Best Matching Unit
 Calculate neighbourhood radius of BMU
 Adjust weights accordingly in neighbourhood
end while

D. Finding the Best Matching Unit

The Best Matching Unit is the neuron with a weight vector that most closely resembles a given sample vector. Finding the BMU can be considered as a competitive learning technique in which each neuron has a chance to win but only the best is selected. The BMU can be found using a variety of techniques [1]. The most popular approach is calculating the distance between the given sample vector and the BMU weight vector. One of the most common distance measures is Euclidean distance, which measures the squared distance between two vectors. Given two vectors, $v_1 = [x_1 \dots x_n]$ and $v_2 = [y_1 \dots y_n]$ where n is the number of elements in the vector, then the distance between the two is measured as:

$$distance = \sqrt{(x_1 - y_1)^2 + \dots + (x_n - y_n)^2} \quad (1)$$

The lower the distance, the closer the two vectors are to one another. If we want to find the BMU, all we do is search the network for the neuron that has a weight vector which has the least amount of distance to the given sample vector.

E. Neighbourhood Update

The neighbourhood of a given neuron constitutes as an abstract area around the unit in which changes in that neuron will effect other neurons within that area. Once the BMU is found, it's neighbourhood is calculated based on some decay function. It can decay linearly, however it's more common to use an exponential decay function similar to equation 2. σ_0 represents the initial neighbourhood radius. A new radius is calculated basic on the decay function with respect to time t and a decay constant λ . This process of updating the weights in a specific neighbourhood can be viewed as a form of co-operative learning, where the neurons work together to more similar to the BMU. [1]

$$\sigma(t) = \sigma_0 e^{-\frac{t}{\lambda}} \quad (2)$$

The initial choice in the radius of the neighbourhood can alter the outcome of the SOM drastically. The radius has to be large enough to ensure that enough neurons in the lattice are getting a chance to fire. If the radius is too small, a sporadic behaviour appears in the map and the direction of ordering for the cluster will change discontinuously. [1] There is a more in depth explanation in the Analysis section on the effects of the initial neighbourhood radius size.

F. Learning Rule

The learning rate for the network work represents how big of factor to apply to the weight update equation when it comes time to adjust a neuron's vector. The learning rate can vary from values ranging from 0.0 ... 0.9. Initially the factor α_0 should be a large value, typically somewhere around 0.7-0.9. Like the neighbourhood radius, this value also decays with respect to time according the the following equation:

$$\alpha(t) = \alpha_0 e^{-t/\lambda} \quad (3)$$

Where $\alpha(t)$ represents a new learning rate at the given time t . By continuously decaying the learning rate, this allows the network to initially makes large changes in the weights to more quickly isolate clusters, then over time, to help the network converge, the small learning rate will help make the minor changes necessary to get the best characteristics from a cluster.

G. Weight Update

Once the BMU is found and the neighbourhood radius is calculated, the new learning rate found above is applied to the following weight update equation:

$$W_{i+1} = W_i + \alpha(t)(V_i - W_i) \quad (4)$$

The new weight is found by adding the old weight plus the learning rate multiplied with the difference between the sample vector V_i and the weight vector W_i . By updating the weights in this manor, the weight vectors of the neighbourhood neurons will update accordingly and make their vectors more similar to the sample vector. Repeating these steps over a number of iterations allows the network to properly learn and classify the data set it's been given.

IV. DATA AND SAMPLES

A. Problem 1 - Colour Organization

The goal of this problem is to present the SOM with randomly coloured pixels and have it classify them in similar groups based on their Red, Green and Blue colour values. An initial image of 200x200 pixels is created as shown in figure 4. The random colors chosen are limited to 8 different colours as shown in table I

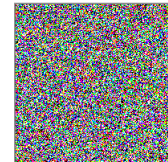


Fig. 4: Random Sample Image - 200x200 pixel colours of image are randomized to get 8 different colours, namely - Red, Magenta, Blue, Cyan, Green, Yellow, Black, White. Each pixel will serve as a data sample.

For every iteration of the SOM algorithm, a pixel is randomly chosen from this image. The sample vector is constructed using the RGB values and it's this vector that will be used to find the Best Matching Unit in the network.

TABLE I: Random Colour Values

Colour	(R,G,B) Value
Red	(255,0,0)
Magenta	(255,0,255)
Blue	(0,0,255)
Cyan	(0,255,255)
Green	(0,255,0)
Yellow	(255,255)
Black	(0,0,0)
White	(255,255,255)

The objective here is to obtain a classification for each colour based on it's corresponding cluster. An example of the map after training is shown in figure 5. Note how clusters that have similar shade in color, like green, yellow and cyan are positioned closer to one another. As opposed to colours with high contrast like black and white, their positions differ greatly on the map.

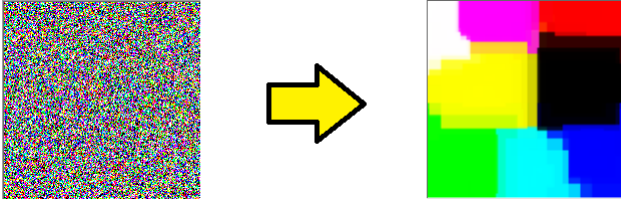


Fig. 5: Random to Organized - The random colours of the picture on the left have been fed through the network and the SOM has grouped them accordingly. Clusters sharing a similar colour are closer together than clusters with varying colours.

B. Problem 2 - Image Clustering

Image clustering works similar to the previous problem, however this time our sample vectors are no longer going to be random RGB values. For image clustering, we want to pass through the network whole images in hopes that it can classify a variety of images based on some unique features. For this analysis a group of images was chosen at random. I chose to use characters from the Super Nintendo game Donkey Kong Country. I chose 7 different characters, as shown in figure 6 from the game and gathered 3 images of each for training data, although just one picture for each character is shown. Note that there are only two images for Krusha, I was unable to find a suitable third picture.

Originally the goal was to have the SOM cluster members of the Kong family, however it was quickly realized that the image quality of each was not significant enough to have the SOM produce a clear cluster distinction. The addition of K. Rool and Krusha allowed the SOM to better organize the

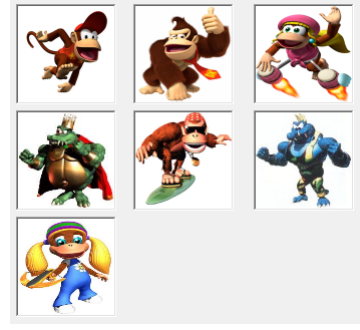


Fig. 6: Sample Images - Sample images used in training. Starting from the top left going right - Diddy Kong, Donkey Kong, Dixie Kong, King K. Rool, Funky Kong, Krusha, Tiny Kong. Note that there are a total of 3 images for each character(excluding Krusha) that are actually used in training.

pictures based on their overall appearance or any hidden data that might be hiding in the higher dimensions.

The sample vector is constructed differently that the previous RGB vector. A vector v_1 is constructed that has the form:

$$v_1 = [\mu(RGB), \sigma(RGB), \mu(R), \sigma(R), \mu(G), \sigma(G), \mu(B), \sigma(B)] \quad (5)$$

Where $\mu(RGB)$ and $\sigma(RGB)$ is the average and standard deviation values respectively for the Red, Green and Blue values for each pixel over the entire image. The rest of the values are corresponding to the average and standard deviation of each Red, Green and Blue color over the entire image.

In order to create a unique classification for each picture, every picture is partitioned into 9 equals parts as shown in figure 7.

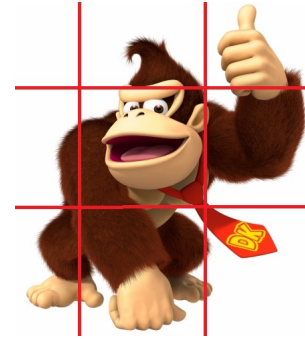


Fig. 7: Example of Image Partition - Every image in the collection is partitioned into 9 equal parts in order to construct a sample vector based on that picture's unique colour qualities.

Another vector v_2 is now constructed using each of the partitions information. v_2 takes the following form:

$$v_2 = v_1 + [\mu(RGB_t), \sigma(RGB_t), \mu(R_t), \sigma(R_t), \mu(G_t), \sigma(G_t), \mu(B_t), \sigma(B_t)]$$

$$v_2 = [\mu(RGB_t), \sigma(RGB_t), \mu(R_t), \sigma(R_t), \mu(G_t), \sigma(G_t), \mu(B_t), \sigma(B_t),$$

$$\sigma(B_t), \mu(RGB), \sigma RGB, \mu(R), \sigma(R), \mu(G), \sigma(G), \mu(B), \sigma(B)] \quad (6)$$

Where $|v_2| = |v_1| + 18 = 8 + 18 = 26$ and where S_i represents the i^{th} subsection of the partitioned image, and $\mu(S_i)$ and $\sigma(S_i)$ are the average and standard deviations of the combined Red, Green and Blue values for each section. This new vector v_2 represents a training vector for the SOM. In total there are 20 sample vectors, each vector representing one of the images respectively.

V. ANALYSIS

In order to have a full understanding of Self-Organizing Maps, it's a good idea to have an understanding of the parameters and the effects they have on the network should they be changed. An analysis was done on the neighbourhood size to show the effects of altering the initial radius size. Analysis was also done on the learning rate, and how changes to the initial learning rate values can effect the map.

Analysis was done using a map size of 200 neurons by 200 neurons and a decay constant of $\lambda = 15.0$. The algorithm ran for a total of 250 iterations. From observation evidence the colour organization seems to converge around that mark, any more iterations and the SOM would be over-fed.

TABLE II: Learning Rate Values

α
0.9
0.7
0.5
0.2

TABLE III: Neighbourhood Radius

σ
200
100
50
20

A. Validity Measures

Since clustering is a unsupervised training process, the evaluation of the Self-Organizing Map after training is crucial in determining the validity of clusters. There are several clustering validity measures, however we will focus on the Dunn validity measure [6]. Introduced by Dunn [7], the Dunn index definition is as follows:

$$D = \min_{i=1..n_c} \left\{ \min_{j=i+1..n_c} \left\{ \frac{d(c_i, c_j)}{\max_{k=1..n_c} \text{diam}(c_k)} \right\} \right\} \quad (7)$$

$$d(c_i, c_j) = \min_{x \in c_i, y \in c_j} \{d(x, y)\} \quad \text{diam}(c_i) = \max_{x, y \in c_i} \{d(x, y)\} \quad (8)$$

Where d is the number of dimensions, $d(x, y)$ is the distance between two data elements, c_i is a given i^{th} cluster and n_c is the number of clusters.

VI. RESULTS

A. Neighbourhood Radius Analysis

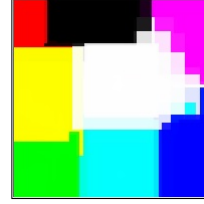


Fig. 8: $\sigma = 200$

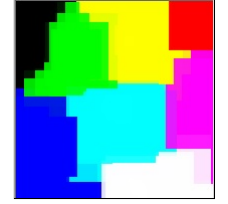


Fig. 9: $\sigma = 100$

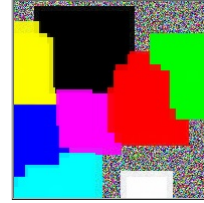


Fig. 10: $\sigma = 50$

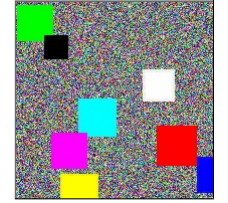


Fig. 11: $\sigma = 20$

Fig. 12: Neighbourhood Radius Analysis

It's quite clear by looking at the final clusters that having a neighbourhood radius size of $\sigma = 50$ and $\sigma = 20$ are not optimal values. With such a small radius compared to the size of the map, not all neurons get a chance to fire, as shown by the random pixel values in map representing the initially random weights. The neurons that are not fired are considered to be "dead neurons" because if they never get a chance to fire in the first initial runs, then it's very unlikely that they will ever get a chance and will most likely be a waste of space. As opposed to a large radius size, where the majority of the network gets a chance to fire in the first few iterations. This clearly leads to a more diverse and tightly knit group of clusters.

B. Learning Rate Analysis

The effect of altering the learning rate is rather impressive. It's clear from the images above that modifying the learning rate has an effect on the granularity of the clusters. There isn't too much difference between having an initial learning rate of $\alpha = 0.9$ and $\alpha = 0.7$, however the change does become more noticeable with smaller values. When $\alpha = 0.5$ the clusters appear more blurry, not quite as crisp as the previous learning rate. This effect is even more elaborated for $\alpha = 0.2$. This would be due to the fact that learning value in the weight update equation 4 is a lot smaller than for other values of α . This implies there will be less of a change in the weight

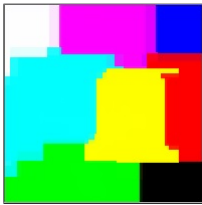
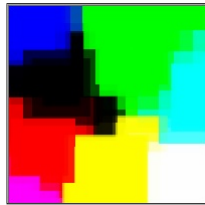
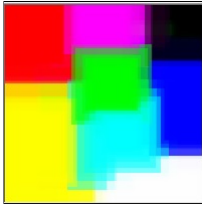
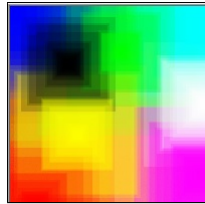
Fig. 13: $\alpha = 0.90$ Fig. 14: $\alpha = 0.70$ Fig. 15: $\alpha = 0.50$ Fig. 16: $\alpha = 0.20$

Fig. 17: Learning Rate Analysis

vector for every neuron in the BMU neighbourhood. In this example this means there will be less change in the color in the neighbourhood and this is reflected in the fuzziness of each cluster. They haven't had a chance to fully learn their own clusters so they are partially merging.

C. Image Clustering

The analysis for Image Clustering is to determine if the Self-Organizing map is capable of organizing and clustering a set of images. Using the entire collection of images shown in figure 6, a sample collection of 20 vector was constructed and fed through the network.

Using the following parameters: $\alpha = 0.7$, $\sigma = 100.0$, $\lambda = 15.0$, Map Dimension: 200x200px, 500 Iterations, the following results were produced over a series of runs as shown in figure 22

The above samples were made by finding the best matching unit for each picture's sample vector after training and displaying their image in their corresponding map position. In all figures it's clear clustering has taken place. Most noticeable would be the King K. Rool images (Green with a red cape and gold crown). The effect of the lighting in each image seems to be affecting the clustering. It's especially noticeable in figure 20, the images on the bottom-left side have a much greater light effect in their overall image. Whereas the images on the top-right area have a much darker appearance. Perhaps with a large training set size it would be possible to have the map clustering strictly on each character's personal colour, such that each character has their own cluster. Another possible solution would be to gather more images with similar lighting and graphics resolution.

VII. CONCLUSION

The colour organization showed very promising results and displayed a clear indication of the behaviour in varying parameters. By adjusting the initial neighbourhood size, we saw the effects it has on updating nearby neurons and would

have to agree with Kohonen [1] when an initial size of at least half the map size is sufficient in obtaining optimal clusters.

The image clustering also showed promising results. We saw how the Self-Organizing map breaks apart images to uniquely classify them in the map and how it orders similar pictures based on this classification. It was clear from the results that similar images were indeed being clustered together. However, based on the quality of each image and the overall uniformity, the clustering algorithm still performed well. We saw how some pictures, even though they were the same Donkey Kong Country character, were being grouped with similar looking pictures that might be due to the brightness or saturation in the picture rather than being grouped with the same character. This might be avoided by choosing a better collection of images or having more training examples of each so the minor differences in each character will stand out more than their image quality and effects.

REFERENCES

- [1] T. Kohonen, *The Self-Organizing Map*, 1990.
- [2] J. Vesanto, *SOM-based data visualization methods*, 1998.
- [3] O. S. T. Kohonen, *Engineering Applications of the Self-Organizing Map*.
- [4] ai junkie, "Kohonen's self organizing feature maps," Accessed March 2013. [Online]. Available: <http://www.ai-junkie.com/ann/som/som1.html>
- [5] D. G. Roussinov, "A scalable self-organizing map algorithm for textual classification: A neural network approach to thesaurus generation," Accessed March 2013. [Online]. Available: http://arizona.openrepository.com/arizona/html/10150/106141/A_Scalable-98.htm
- [6] A. B. Ferenc Kovacs, Csaba Legany, *Cluster Validity Measurement Techniques*, 2005.
- [7] J. C. Dunn, J. C. Dunn: *Well Separated Clusters and Optimal Fuzzy Partitions*, *Journal of Cybernetics*, Vol 4., 1974.
- [8] E. A. Juha Vesanto, *Clustering of the Self-Organizing Map*, 2005.

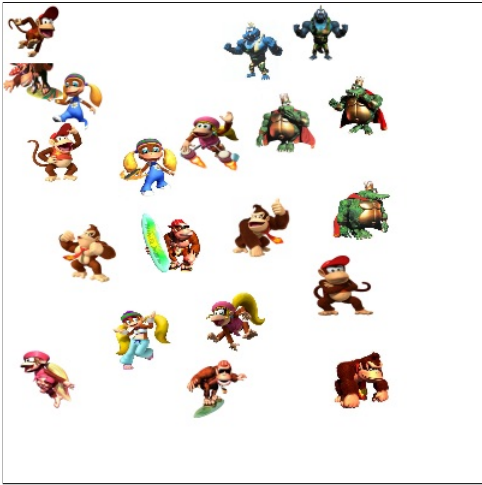


Fig. 18: Result 1

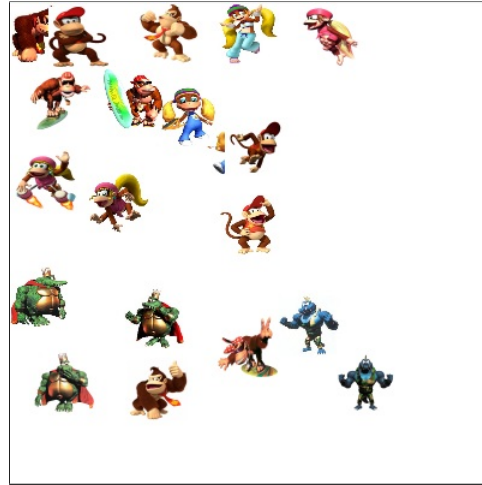


Fig. 19: Result 2

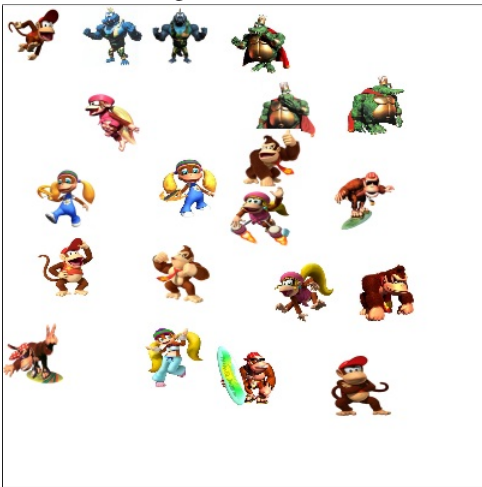


Fig. 20: Result 3

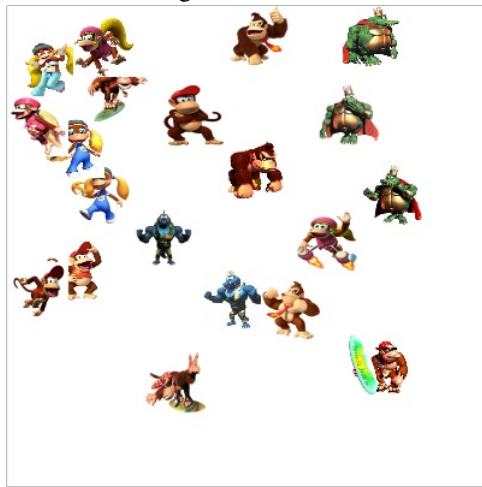


Fig. 21: Result 4

Fig. 22: Image Clustering Results - 20 image sample collection. Similar looking images are grouped together. Images with a large difference in appearance are generally farther apart.