

PROCESO DE RECONOCIMIENTO DE OBJETOS, ASISTIDO POR COMPUTADOR (Visión Artificial), APLICANDO GASES NEURONALES Y TÉCNICAS DE MINERÍA DE DATOS

RESUMEN

El presente trabajo desarrolla un modelo de reconocimiento de objetos basado en la forma de los mismos. Para dicho fin se revisan y comparan muchas técnicas de Inteligencia Artificial que son potenciales herramientas para resolver el problema; demostrando así la efectividad única que posee para dicho fin una de las mejores técnicas para cluster en IA, desarrollada entre otros, en trabajos doctorales de la Universidad de Alicante y Valladolid (España), como lo son los gases neuronales crecientes. Finalmente se enuncian las posibilidades tangibles que esta técnica abre no sólo para reconocer formas, sino también, texturas, colores y movimiento (visión completa).

PALABRAS CLAVES: Gases Neuronales, C++, reconocimiento, Visión Artificial.

ABSTRACT

The present work develops a model of recognition of objects based on its form. For this, much techniques of Artificial Intelligence that are potential tools to solve the problem are studied and compared; demonstrating the best answer given for the most recent tendencies in IA, developed mainly in doctoral works of the University of Alicante and Valladolid (Spain), like the growing neurals gases. Finally it is showed in the article that this technique is not just useful to recognize the forms, but it can recognize textures, colors and movement too.

KEYWORDS: *Growing neural gas, recognition, Artificial vision.*

1. INTRODUCCIÓN

En la desesperada búsqueda que en los últimos años ha emprendido el hombre (ingeniero), para desarrollar sistemas computacionales basados en modelos de otras áreas del conocimiento como la medicina (biológicos) bajo el nombre de una disciplina llamada Inteligencia Artificial [3]; se ha encontrado con miles de retos al desarrollar algoritmos humanizados como las redes neuronales, sistemas expertos, algoritmos genéticos etc. Pero ningún reto más motivador, encantador y exigente, como el de sustituir el ojo humano por una maquina computacional, o por lo menos dotar del sentido de la visión (el más complejo) dicha maquina.

El desarrollo de diferentes técnicas con este fin alrededor del mundo, sigue evolucionando de forma exponencial, más sin embargo, es un hecho que aún hay mucho camino por recorrer, para llegar a concebir un sistema 100% eficiente en este campo, y que involucre toda la gama de posibilidades y complejidades que el mismo conlleva. Puesto que no hace falta enumerar las dificultades que acarrea el intento de generar un sistema de visualización computarizada perfecto, y en el momento podemos simplemente asumir que son demasiadas, ya que tampoco pretendemos solucionarlas todas en este trabajo. Es necesario enfocarse en tratar de resolver una necesidad básica, primordial y altamente compleja de dicho sistema, como lo es, aprender a

reconocer formas de objetos para poder clasificarlos correctamente. Tratando de desarrollar esta solución

mediante una técnica que sea eficiente, confiable y que no se desmesure constante u ocasionalmente en el costo computacional (sistema estable).

2. PLANTEAMIENTO DEL PROBLEMA

El problema de reconocer la forma de un objeto para su debida clasificación, por medio de un computador, se puede solucionar, o por lo menos así se intentó solucionar en el presente trabajo, mediante la aplicación de la siguiente metodología (algoritmo global):

1. Fotografiar el objeto (no blanco), estático, en fondo blanco, con buena luz, y totalmente solo (sin sombras).
2. Escalar la foto a blanco y negro, mediante un algoritmo computacional discriminante, cuyo objetivo es: todo píxel de la foto de color blanco o casi blanco lo vuelva negro y cualquier otro píxel lo convierta a blanco (umbralización).
3. Considerar la parte blanca de la foto como un subespacio de entrada, de R^2 , tal que alguna técnica de IA se adapte topologicamente a él. (segmentación).

DELIA JOHANNA MUNOZ.

Estudiante

Universidad Tecnológica de
Pereira.

delia.johanna@gmail.com

4. Luego que la técnica usada puede representar la topología del espacio de entrada (la forma del objeto). Hallar los patrones y características que este posee, para tratar de clasificarlo con una técnica de data mining como es el clustering o agrupamiento (mediante cluster's naturales).

Nótese que el meollo del problema está en encontrar la técnica que se adapte topológicamente a un subespacio de entrada para poder representarlo. Razón por la cual este trabajo se dedica a continuación, exclusivamente a dicho propósito.

3. POTENCIALES HERRAMIENTAS

Las siguientes técnicas neuronales de inteligencia artificial, que se van a enunciar, utilizadas comúnmente en data mining, son denominadas auto-organizadas y no supervisadas, esto quiere decir grosso modo que poseen elementos que se adaptan o tratan de imitar el espacio que se introduce a la entrada patrón a patrón. Por lo tanto el sistema a partir de un proceso de autoorganización proporcionará cierto resultado, el cual será el reflejo de las relaciones de similitud existente entre dichos patrones de entrada [2].

3.1 Redes neuronales de kohonen

Podríamos referirnos a los mapas autoorganizados de Kohonen [1], como la técnica de claustrización más usada. Esta red se compone de dos capas de neuronas una lineal y otra matricial, figura 1, la primera de ellas (capa de entrada), tiene tantas neuronas como dimensiones tenga el espacio de entrada, y la segunda, que se ajusta después del entrenamiento a los patrones de entrada, tiene tantas neuronas como deseemos, tal que, la adaptación sea total o por lo menos abarquemos la mayor parte del espacio de entrada (cantidad de patrones de entrenamiento). El proceso de entrenamiento de esta red es competitivo [1], pero no es preciso detallarlo en este momento, en verdad es más importante revisar de que manera puede resolver nuestro problema.

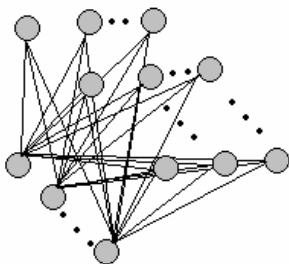


Figura 1. Estructura Típica de un mapa de Kohonen.

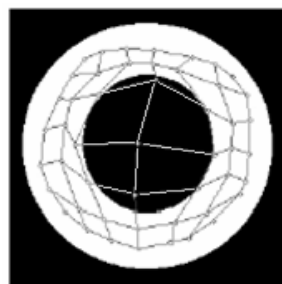


Figura 2. Ubicación final de las neuronas, con un espacio de entrada en forma de anillo.

Se utiliza entonces para una prueba, como espacio de entrada, un subespacio de R^2 el cual se compone de todos los puntos blancos de la figura 2, (no todos, pero si un gran numero), lo cual supone que nuestra red posee 2 neuronas en la capa de entrada y un numero suficiente de neuronas en la capa de salida, luego de un entrenamiento exhaustivo de la red, finalmente las neuronas de la capa de salida, quedan ubicadas (autoorganizadas) y conectadas, tal y como se ve en la figura 2.

Estas redes son comúnmente utilizadas para reducción de dimensiones, reconocimiento de texto, reconocimiento de voz, cuantificación vectorial... Y podemos observar que son también especialmente prácticas para nuestro cometido, ya que se adaptan de manera aceptable a nuestro subespacio de entrada, como lo muestra la figura 2. El cual sería, en nuestro caso la silueta (rellena), del objeto a reconocer.

3.2 Growing cell structures

Este modelo posee una estructura flexible, con un número variable de neuronas, y una topología k-dimensional donde k es escogido arbitrariamente y a priori. Las neuronas están conectadas entra sí, formando hipertetraedros. En el caso más usual en el que la red es bidimensional, las neuronas están unidas formando triángulos [4].

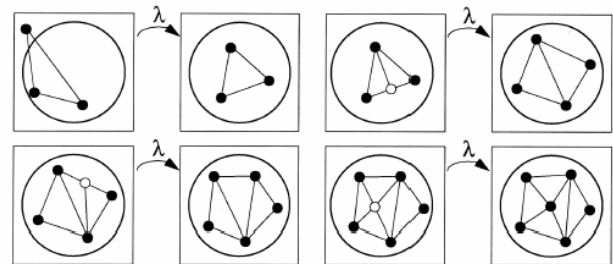


Figura 3. Acomodación e inserción de neuronas mediante el proceso de entrenamiento del growing cell structure.

Tampoco podemos detallar en el proceso de entrenamiento de la red, ya que entre otras cosas, es común [4]. Es suficiente con decir, que inicia con una cantidad determinada de neuronas y dependiendo de un error que calcula con los patrones de entrada, va acomodando e insertando nuevas neuronas para reducir dicho error. Algo muy importante y que la hace más optima que la red de Kohonen, es el hecho que puede formar subredes totalmente independientes.

La red reaccionó a la misma prueba de anillo que se le hizo a nuestra anterior red, de la forma que se observa en la figura 4.



Figura 4. Ubicación final de las neuronas, con un espacio de entrada en forma de anillo

3.3 Gases neuronales

Indudablemente esta técnica que apenas surge en el campo del desarrollo en sistemas de inteligencia artificial, además de innovadora, es demasiado efectiva, razón por la cual se hace mayor hincapié en ella, además también por haber sido la más apta para la solución. Este modelo abarca varias características de otros modelos autoorganizados como [4]:

Realizan su proceso de adaptación según una función de energía (espacio de los vectores de entrada) a diferencia de Kohonen, en el que viene determinado por la estructura de la red.

Convergen rápidamente a errores de cuantización pequeños, menores que los obtenidos con los modelos anteriores.

Estos modelos pretenden evitar la restricción de los modelos anteriores, donde se requiere un conocimiento a priori sobre la dimensión topológica del espacio de los vectores de entrada. Por ello no preestablecen ninguna topología de red, sino que es durante el aprendizaje cuando las neuronas se conectan o desconectan reajustando su red de interconexión.

3.3.1 Neural Gas

Este modelo (simple) posee todas las características mencionadas anteriormente, pero posee la gran desventaja de tener que predeterminedar el tamaño de la red, o sea el número de neuronas con que este contará. Ha sido empleada en la predicción de series temporales, control de robots, clasificación de escrituras.

La mejor manera de ver su evolución o comportamiento es viendo la figura 5, en donde el gas neuronal simple, se adapta o toma la forma de una estructura en tres dimensiones, después de su entrenamiento.

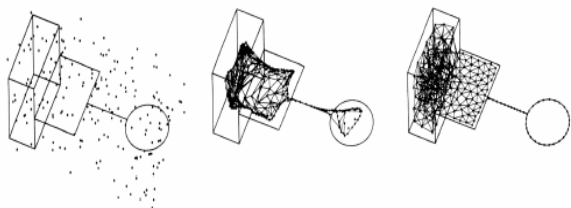


Figura 5. Adaptación de un gas neuronal a una forma de tres dimensiones.

Ahora, el método de entrenamiento de este gas si es de nuestra incumbencia pero no lo esbozaremos por dos razones: Primero, se le debe hacer un detenido estudio que nos llevaría otro artículo entero, segundo se esbozará más adelante uno casi igual, que pertenece a la siguiente técnica a tratar.

3.3.2 Growing Neural Gas

Por fin una técnica que satisface todas nuestras necesidades, necesidades que explicaremos más adelante.

Valiéndose de las bondades del neural gas y de la growing cell structure, surge el gas neuronal creciente, en el que no se establece topología de unión entre las neuronas y tampoco el número de neuronas de la red, sino que a partir de dos neuronas, el gas se va reproduciendo y ubicando, generando y uniendo neuronas hasta que toma la forma exacta del espacio de entrada, y a nuestro parecer esta es la mayor flexibilidad que se puede obtener en un sistema, además de la rapidez y disminución en error a comparación con otras técnicas.

Este modelo es utilizado comúnmente en la robótica e identificación de gestos, y en la figura No 6 podemos ver su evolución para el mismo cuerpo tridimensional como espacio de entrada.

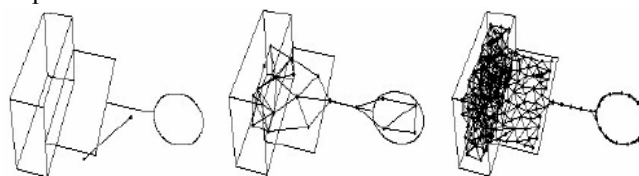


Figura 6. Adaptación de un gas neuronal creciente a una forma de tres dimensiones.

Luego de ver estos dos métodos en acción, es justo observar su comportamiento para el ejercicio que veníamos haciendo con los anteriores modelos (anillo), figuras No 7,8. Podrán notar que lo hace mejor el gas neuronal simple pero más adelante demostraremos porqué de manera global, es mejor y más útil el gas creciente. Para finalizar esta sección esbozaremos el algoritmo en pseudocódigo y totalmente en español (difícil de encontrar) para un gas neuronal creciente.

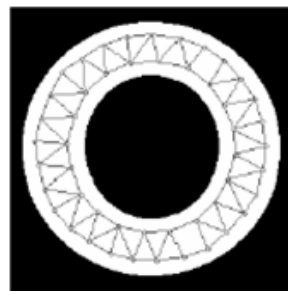


Figura 7. Ubicación final de las neuronas para un Neural Gas.

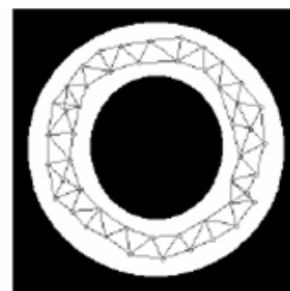


Figura 8. Ubicación final de las neuronas para un growing neural gas.

3.3.3 Algoritmo de entrenamiento en pseudocódigo para un Growing Neural Gas

- **Inicio:** Cree 2 neuronas de gas, aleatoriamente sobre la dimensión a la que pertenece el espacio de entrada (R^n) inícielas con error local acumulado (error k) igual a 0, únalas con conexión de edad igualmente en 0.
- Genere una señal de entrada $X \in (R^n)$, que conforma la distribución del espacio de entrada.
- Localice las dos neuronas (s,t) mas parecidas a la señal X de entrada. O sea aquellas 2 neuronas con vectores de pesos asociados W_s y W_t , cuya distancia euclidiana con X, $\|W_s-X\|^2$ y $\|W_t-X\|^2$, sea la mas pequeña y la segunda más pequeña respectivamente, de las k neuronas que hallan en ese momento.
- La neurona ganadora (s) debe actualizar su error local, para lo cual le sumaremos a este la distancia euclidiana entre el vector de entrada y su vector de peso:

$$error_s = error_s + \|W_s-X\|^2$$
- Mueva la neurona ganadora s, y a todos sus vecinas (todas aquellas que tengan conexión con s) hacia la señal de entrada X, en factores de sus distancias de e_s y e_n así ($e_s, e_n \in [0,1]$):

$$W_s = W_s + e_s(X - W_s)$$

$$W_n = W_n + e_n(X - W_n)$$

Para todo n que sea vecina de s

- Incremente la edad de todas las conexiones de la neurona s con sus vecinas topológicas.
- Si la neurona s y t están conectadas ponga a esa conexión edad 0, si no existe pues entonces créela.
- Si hay alguna conexión con edad mayor a a_{max} , destrúyala, si después de esto alguna neurona esta total mente desconectada también elimínela.
- Si la iteración actual es un múltiplo entero de λ , y la cantidad máxima de nodos aun no se ha alcanzado, entonces inserte una nueva neurona r, así:
 1. busque la neurona u, con el error local acumulado mas grande
 2. Entre los vecinos de u, busque la neurona v con el error mas grande
 3. Inserte la nueva neurona r entre u y v así:

$$W_r = (W_u + W_v) / 2.$$
- 4. Cree conexiones entre u y r y entre v y r, retire la conexión entre u y v.
- 5. Disminuya el error de u y v y ponga el error local de r:

$$error_u = \alpha error_u$$

$$error_v = \alpha error_v$$

$$error_r = error_u$$
- Disminuya el error de todas la neurona s j, por un factor β :

$$error_j = error_j - \beta error_j$$

- Si el criterio de parada aun no se da entonces repita el algoritmo, un criterio puede ser el numero máximo de nodos.

4. ¿CUAL ES LA MEJOR TECNICA?

El mejor criterio de evaluación para las técnicas expuestas se basa en dos requisitos: Primero la técnica que mejor preserve la topología [4], y segundo, teniendo en cuenta que el sistema es un sistema en tiempo real, también es vital el coste computacional de la técnica o el consumo de tiempo que esta requiera para converger. Por lo cual se debe realizar un balance entre estos dos ítems y decidir cual de las técnicas proporciona mejor desempeño. Para este fin, podemos sustentarnos en la siguiente tabla (1), que resume los entrenamientos y pruebas a las que se sometieron las técnicas.

	Tiempo(seg) ESTRELLA	Tiempo(seg) CIRCULO	Tiempo(seg) ESPIRAL
Kohonen	36	28	42
Cell Structure	12	7	21
Neural Gas	109	95	148
G. Neural Gas	14	8	25

Tabla 1. Resultados de algunas pruebas para todas las técnicas expuestas.

Concluamos entonces: Que para todas las figuras anteriores o espacios de entrada (estrella, círculo...) y otros adicionales definitivamente el más veloz de todos los métodos, fue el de growing cell structures, pero como vimos en el caso de los anillos no es el que mejor se adapta o conserva la topología (aunque no lo hace mal), Podemos verificar sin duda para todos los casos, incluyendo el anillo, que el método que mejor preserve la topología o se adapta es el del gas neuronal simple, pero también es evidentemente, que es el que más tiempo consume en todos los casos. Así que si observamos el método de growing neural gas o gas neuronal creciente, este está totalmente balanceado y es el optimo, ya que fue el segundo método que mejor preservó la topología, y también el segundo método que consumió menos tiempo (aunque casi igual al mejor). Por este motivo, y la secuencia de pruebas adicionales que se le hicieron en tiempo real y con figuras o espacios complejos, concluimos que este es el mejor método para representar topológicamente un espacio de entrada en tiempo real.

5. PASO A SEGUIR: DEL GAS AL GRAFO

Es evidente que después del entrenamiento neuronal, lo que obtenemos es un complejo grafo (nodos=neuronas, aristas=conexiones), que tiene la forma de nuestro objeto. Pero momentáneamente, es solo importante para nosotros, la silueta del objeto (para reconocer su forma) y podemos obviar el interior o relleno del grafo, para

quedarnos simplemente con un grafo de contorno del grafo original. Para esto utilizamos un algoritmo simple que cumple con el siguiente lema:

Dado un grafo, se define como su grafo de contorno el conjunto de aristas que pertenecen únicamente a un polígono (las demás se destruyen, al igual que los nodos totalmente desconectados). La aplicación de esta técnica para obtener el grafo silueta se muestra en la figura No 9.

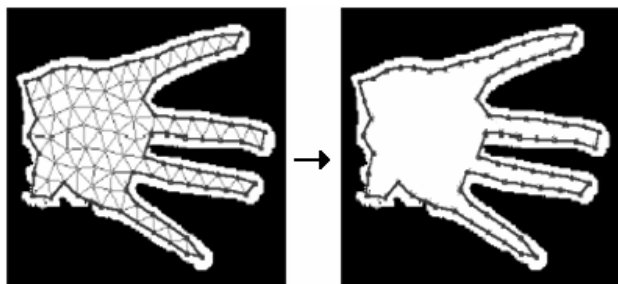


Figura 9. Obtención del grafo de contorno a partir del grafo neuronal.

Después de obtener este grafo de contorno podemos hacerle más tratamiento como reducción de nodos con detección de esquinas (teoría de grafos). Sin embargo lo que en verdad nos interesa es que este grafo guardará características únicas de su forma (la forma del objeto), tal que nos permitirán distinguirlo o clasificarlo agrupadamente (con data mining “clustering”) con grafos que tengan características similares (objetos de la misma especie o con la misma forma). Algunas características se enuncian a continuación para un grafo, de manera resumida (extracción de características)[4]:

Dado un grafo de contorno, el diámetro se define como la distancia máxima existente entre dos vértices (neuronas) cualesquiera del mismo. Esta medida es invariante con rotaciones o traslaciones.

Dado un grafo de contorno, se define su longitud como la suma de las longitudes de todas las aristas que le conforman. Esta medida es igualmente invariante a rotación o traslación.

La Curvatura de un grafo de contorno, viene dada por el ritmo de la variación de las pendientes de sus aristas, comenzando siempre desde el mismo punto. Con esta característica se pueden hallar las esquinas del grafo y los segmentos rectos. En la figura 10, se muestra la función de curvatura para varios elementos de dos clases, puede verse que cada clase conserva un patrón similar de curva.

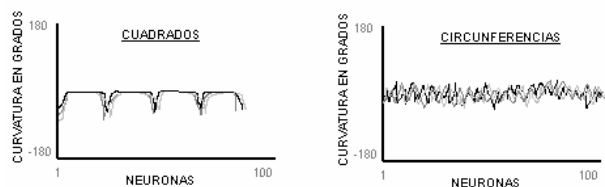


Figura No 10. Función de curvatura para varios cuadros y varias circunferencias.

La Firma del grafo, se obtiene calculando la distancia de cada uno de los vértices al centro del grafo, Igual que en la curvatura se debe partir del mismo punto para todos los grafos. Si se divide esta medida por la distancia entre el nodo más lejano y el centro, se obtiene una medida invariante a al rotación y traslación.

Los Códigos de Cadena se emplean para representar un contorno mediante una sucesión de segmentos de dirección especificada. Usualmente se basa en segmentos de conectividad cuatro u ocho. En la figura 11, se muestra la función de código de cadena para varios elementos de dos clases, empleando conectividad ocho.

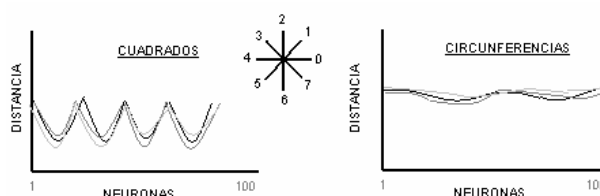


Figura 11. Función de código de cadena para varios cuadros y varias circunferencias.

Existe más variedad de características de grafos como los momentos, pero en aras de resumir, es suficiente con lo expuesto hasta el momento.

6. ÚLTIMO, PROCESO DE CLASIFICACIÓN

Luego de obtener el respectivo grafo de contorno de un objeto, con toda la respectiva información que este nos puede brindar, información que es exclusiva de cada grafo y que debe guardar un alto grado de similitud, con grafos parecidos o grafos que representan objetos con igual forma. Nos valemos de esto para clasificarlos o agruparlos, Creando un sistema de clustering, que al recibir la información de un nuevo grafo, le clasificará en un grupo (manzanas, bananos, estrellas...) basándose en la similitud de información que el grafo tenga con este.

Para este proceso de clasificación, que es multidimensional, en donde inclusive se deben tratar de agrupar funciones con características similares, es recomendable utilizar una convencional técnica de minería de datos [5] o data mining, que cumpla con dicho propósito. Por lo tanto se ha implementado una red neuronal autoorganizada de Kohonen, de igual forma se pudo utilizar nuevamente un gas neuronal, pero en aras de la versatilidad y merma en la complejidad del sistema, se opto por la primera opción. Este mapa de Kohonen, que se implementa es de alta dimensionalidad, ya que los patrones a clasificar (grafos) a sí lo son. Puesto que un patrón P (grafo), no se componen de dos o tres dimensiones, sino, de varias funciones $f(x)$, $g(x)$... y

números reales a, b, \dots . Como su curvatura y su diámetro etc. ($P[f(x), g(x), \dots, a, b, \dots]$)

7. CONCLUSIONES

Hemos querido que las conclusiones sean propias y se basen en la figura 12, la cual representa resumidamente los resultados de un ejemplo de clasificación del sistema, para varios objetos reales de diferentes clases.

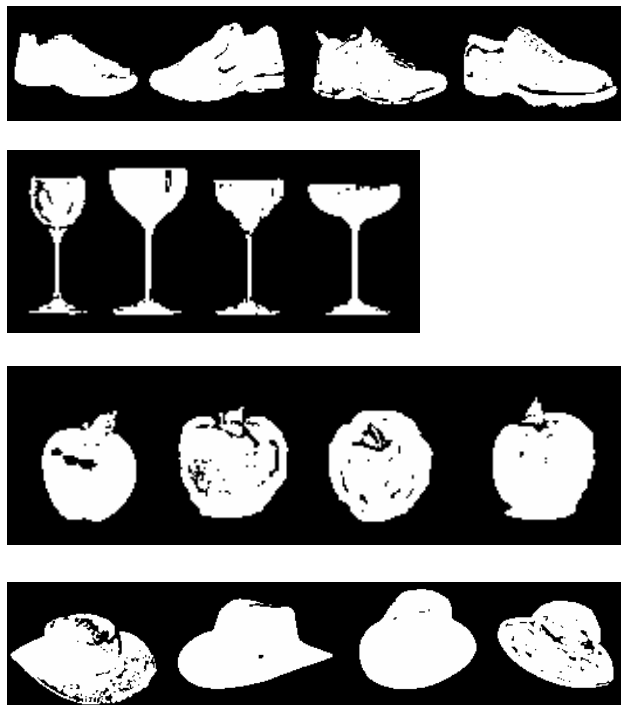


Figura 12. Modo en que el sistema clasificó diferentes elementos introducidos aleatoriamente.

La figura 12, muestra pues los cuatro grupos que el sistema formó después de haberle introducido fotos de aleatoriamente de diferentes tipos de zapatos, copas, manzanas y sombreros. Cabe resaltar que estos mismos grupos se generaron después de ingresar los elementos varias veces en órdenes diferentes y aleatorios.

Es importante en este momento aclarar, que la técnica de los gases neuronales no llega a dar su máximo potencial en este ejemplo de reconocimiento y clasificación basado en formas, puesto que no debemos olvidar que de los resultados de acoplamiento a la topología o espacio de entrada, que resultaron ser un grafo, han sido inexplorados en un 90%, ya que eliminamos todo el interior del grafo para obtener el grafo de contorno. Pero en verdad este grafo completo, genera mucha más información de la que pensamos y al hacer un tratamiento especial de esta, podemos obtener características de texturas, sombras, colores... Inclusive con secuenciación de entrenamientos a ráfagas de fotos en tiempo real se puede obtener información, sobre una escena, y los movimientos que hay en ella. En fin este estudio es

apenas una pequeña muestra de la capacidad de esta técnica.

Para finalizar se muestran en la figura 13 las fotos reales de los elementos clasificados por el sistema, con el fin de observar en detalle el buen desempeño del mismo.



Figura No 13. Fotos originales de todos los elementos que fueron clasificados por el sistema, de la manera que se muestra en la figura No 12.

8. AGRADECIMIENTOS

El autor brinda sus agradecimientos, al ingeniero Juan Diego Gómez (UTP), quien dirigió el proyecto y colaboró en la programación de software y diseño de algoritmos. A sus padres Luís Alfonso M y Elizabeth V, por su apoyo y cariño incondicional.

9. BIBLIOGRAFÍA

- [1] B.M del Brio, "Redes Neuronales y sistemas difusos" 2da ed. Vol 1. Ed. Zaragoza España: Alfaomega Ra-Ma 2002.
- [2] J.R Hilera "Redes Neuronales Artificiales" Ira ed. Vol 1. Ed Madrid España: Alfaomega Ra-Ma 2000.
- [3] Nilson "Inteligencia artificial" Ira ed. Vol 1. Ed Madrid España: Alfaomega Ra-Ma 2000.
- [4] Ph.d Revuelta "Modelo de representación y procesamiento de movimiento en tiempo real" Universidad de Alicante. España, 2001.
- [5] Orallo "introducción a la minería de datos" Ira ed. Vol 1. Ed Madrid España 2004: Pearson (Prentice Hall).