

# OVMMSOM: Una variación de MMSOM y VMSOM como técnica de clusterización de imágenes

Franco Sánchez Huertas  
Escuela de Ciencia de la Computación  
Universidad Católica San Pablo  
Arequipa, Perú  
franco.sanchez@ucsp.edu.pe

Yván J. Túpac Valdivia  
CI en Ciencia de la Computación  
Universidad Católica San Pablo  
Arequipa, Perú  
ytupac@ucsp.edu.pe

Raquel Patiño Escarcina  
Escuela de Ciencia de la Computación  
Universidad Católica San Pablo  
Arequipa, Perú  
rpatino@ucsp.edu.pe

**Resumen**—Este artículo propone el modelo *Optimized Vector and Marginal Median Self-Organizing Map* (OVMMSOM) como una nueva forma de clusterización, como una variante de los Mapas Auto-Organizados (SOM) basado en estadísticas de orden, *Marginal Median SOM* (MMSOM) y *Vector Mediana SOM* (VMSOM). Este modelo emplea combinadamente MMSOM y VMSOM definiendo la importancia de cada uno a través de un índice de participación representado por un parámetro  $\lambda$ . Para demostrar la efectividad de la propuesta se probó con imágenes del conjunto de datos COIL100 y el índice de validación CDbw. Los experimentos realizados mostraron que el modelo propuesto supera los resultados obtenidos por una red SOM estándar entrenada en *batch*, e incluso a los entrenamientos realizados por MMSOM y VMSOM por separado.

**Keywords**—SOM ; MMSOM ; VMSOM; CDbw

## I. INTRODUCCIÓN

Los mapas auto-organizados (*Self-Organizing Maps*, SOM) son un tipo de Redes Neuronales[1] que establecen la adaptación de cualquier espacio de entrada de datos hacia una red de dos o tres dimensiones, de modo que se tenga un prototipo de neuronas bien definidas. Los nodos están organizados en un mapa y compiten entre sí para ganar uno a uno los patrones de entrada [2].

Este modelo está basado en la clase de algoritmos de entrenamiento SOM que emplean estadísticas de orden, como *Marginal Median SOM* (MMSOM) y *Vector Median SOM* (VMSOM)[3]. Estas variantes de SOM son tan buenas como una SOM estándar, entrenada por *batch*, y han sido aplicadas anteriormente al reconocimiento de patrones. La contribución de esta investigación está en la evaluación del trabajo de estos dos algoritmos de entrenamiento juntos mediante un índice de participación representado por  $\lambda$ . Este índice será elegido dependiendo del mejor valor que obtenga el índice *Composity Density between and within clusters index* (CDbw) durante las pruebas. La superioridad de este modelo frente a BL-SOM y sus padres MMSOM y VMSOM queda demostrada mediante los experimentos realizados sobre el conjunto de datos de COIL100 vectorizados con el descriptor *Color Layout* [4].

Este artículo se ha desarrollado de la siguiente manera. En la Sección 2 se explica el concepto de *Clustering* y el índice

de validación CDbw. La Sección 3 desarrolla los conceptos básicos de la red SOM y sus respectivas variantes. Mientras que en la Sección 4, se describe la propuesta del modelo. Los experimentos y los resultados se presentan en la Sección 5. La Sección 6 muestra algunas consideraciones. Y finalmente la Sección 7, las conclusiones.

## II. CLUSTERING

*Clustering* es el proceso de agrupamiento que permite organizar un conjunto de datos dentro de subconjuntos a manera de grupos, así los datos que están en un mismo clúster son más similares que aquellos que se encuentran en otro clúster[5]. A diferencia de la clasificación, este es un aprendizaje no supervisado, es decir, aprende por observación de ejemplos. Otra característica del clustering es que no existen etiquetas predefinidas de las distintas clases posibles para los datos; por lo tanto, a manera de que el algoritmo se ejecuta los datos progresivamente se identifican con los clusters que van apareciendo[6]. Para que un cluster pueda ser tomado en cuenta y validado debe cumplir dos propiedades elementales: baja similitud inter-cluster y alta similitud intra-cluster[6].

### II-1. El índice CDbw

Con la necesidad de encontrar un criterio que permita determinar si el agrupamiento final realizado es el más óptimo, es decir clusters bien compactos y al mismo tiempo bien separados, se aplica un índice conocido como índice de validación. Wu y Chow presentan un índice de validación relativo que se basa en la densidad inter-cluster e intra-cluster. Este índice es conocido como *CDbw* (*Composity Density Between and within clusters*) y encontrará la relación que hay entre los clusters en función de los elementos más representativos de cada uno. Una vez agrupados los datos en  $c$  clusters se utiliza el índice de validación *CDbw*. Entre mayor sea el valor del índice este indica que es más óptimo. Para calcularlo se utiliza la Ecuación 1, en la cual  $\text{intra\_den}(c)$  es la densidad intra-cluster y  $\text{Sep}(c)$  es la separación entre clusters[7], [8].

$$\text{CDbw}(c) = \text{intra\_den}(c) \cdot \text{Sep}(c) \quad (1)$$

En la densidad intra-cluster definida en la Ecuación 2, el término  $\text{density}(v_{ij}) = \sum_{l=1}^{n_j} f(x_l, v_{ij})$ , en el cual  $x_l$  pertenece al cluster  $i$ , y  $v_{ij}$  es el  $j$ -ésimo punto de representación [7].

$$\text{intra\_den}(c) = \frac{1}{c} \sum_{i=1}^c \sum_{j=1}^{r_i} \text{density}(v_{ij}), c > 1 \quad (2)$$

La Ecuación  $f(x_i, v_{ij})$  se detalla a continuación:

$$w_i(t+1) = \begin{cases} 1 & \text{si } \|x_i - v_{ij}\| \leq \sigma, \\ 0 & \text{en otro caso} \end{cases} \quad (3)$$

donde la desviación estándar promedio  $\bar{\sigma}$  es definida por la Ecuación 4:

$$\bar{\sigma} = \sqrt{\sum_{k=1}^{n_i} \|\sigma(i)^2\| / c} \quad (4)$$

donde el término  $n_i$  representa el número de puntos, y  $\sigma(i)$  es un vector que contiene las desviaciones estándar de los puntos de representación del cluster  $i$ .

Para definir el vector de desviaciones estándar del cluster  $i$  se considera el vector de puntos de representación  $\vec{v}_i = (v_{i1}; v_{i2}; \dots; v_{ir_i})$  con tamaño  $r_i$ . En la Ecuación 5 vemos el  $p$ -ésimo elemento de  $\sigma(i)$  [7].

$$\sigma^p(i) = \sqrt{\sum_{k=1}^{n_i} (x_k^p - m_i^p)^2 / (n_i - 1)} \quad (5)$$

donde  $x_k^p$  son los datos y  $m_i^p$  es la media de cluster  $i$ .

La expresión  $\text{Sep}(c)$  de la Ecuación 1 representa la separación que hay entre los clusters. Para calcular la separación se utiliza la Ecuación 6. Los términos  $\text{close\_rep}(i)$  y  $\text{close\_rep}(j)$  son el par de puntos representativos más cercanos entre el cluster  $i$  y el cluster  $j$ .

$$\text{Sep}(c) = \sum_{i=1}^c \sum_{j=1}^c \frac{\|\text{close\_rep}(i) - \text{close\_rep}(j)\|}{1 + \text{inter\_den}(c)} \quad (6)$$

donde la densidad  $\text{inter\_den}(c)$  se obtiene con la Ecuación 7

$$\text{inter\_den}(c) = \sum_{i=1}^c \sum_{j=1}^c \frac{\|\text{close\_rep}(i) - \text{close\_rep}(j)\|}{\|\sigma(i)\| + \|\sigma(j)\|} \quad (7)$$

cuando  $c > 1$

### III. MAPAS AUTO-ORGANIZADOS (SOM) Y SUS VARIANTES

Las redes neuronales artificiales simulan el comportamiento de las neuronas biológicas y son representadas mediante modelos matemáticos. Si bien es cierto todavía no se ha podido comprender el cerebro humano en su totalidad, se toman las propiedades básicas de las neuronas, tales como la propagación del conocimiento entre las que se encuentran en una vecindad, donde las neuronas más próximas reciben mayor estímulo de lo aprendido por alguna neurona. Los mapas auto-organizados (SOM) son redes neuronales no supervisadas bastante utilizadas en el área de minería de datos.[2], [9].

Para entender el entrenamiento de una red SOM debemos considerar el vector de entrada  $X(t)$ , donde todos los datos

de entrada afectan a cada neurona  $i$ . Según los pesos de las neuronas  $w_i \doteq [w_{i1}, w_{i2}, \dots, w_{in}]$  se determinará una neurona ganadora, por lo que todos los pesos de las vecinas serán afectados y actualizados según la ecuación de aprendizaje 8:

$$w_i(t+1) = \begin{cases} w_i(t) + \varepsilon(t)h_{ic}(t)(x_t - w_i), & i \in N_c \\ w_i(t) & i \notin N_c \end{cases} \quad (8)$$

#### III-A. SL-SOM

El método *Self-Organized SOM* (SL-SOM) complementa el trabajo de la SOM, al segmentar el mapa y calcular el particionamiento más óptimo. Una vez entrenada la red SOM se calcula una matriz de distancias inter-neuronal denominada U-Matrix, la cual es segmentada a partir de un umbral definido. Todas aquellas posiciones de la U-Matrix que tengan una distancia menor al umbral se les asigna un valor 1 y a las mayores un valor 0. De esta manera se genera una matriz binaria [10] en la que los sectores donde se acumulan valores 1 se consideran posibles *clusters*. El problema que se encuentra en este paso, es que hay regiones que no pertenecen a ningún grupo, donde encontramos valores 0. Estas regiones no ayudan a visualizar la segmentación, por eso conviene aplicar un algoritmo de crecimiento de regiones como el Watershed [9], que elimina los valores 0 y redefine los *clusters* encontrados. El algoritmo tradicional de SL-SOM determina mediante un dendograma cuál es el mejor particionamiento. En el dendograma se puede observar cuál es el resultado que prevalece, por su continuidad y estabilidad. Es decir, aquel resultado que da con más frecuencia es el considerado como más óptimo. Para crear el dendograma, es necesario realizar un cálculo del número de *clusters* que se encuentran con distintos umbrales. Aunque este método es subjetivo, nos proporciona un criterio para escoger un buen resultado.

#### III-B. BL-SOM

BL-SOM forma un mapeamiento no-lineal de un espacio de entrada  $d$ -dimensional arbitrario hacia un conjunto de nodos de dos o tres dimensiones (el mapa en sí). Cada nodo está asociado con un vector de pesos  $w = (w_1, w_2, \dots, w_D)^T$  en el espacio de entrada. La BL-SOM es entrenada iterativamente y aprende los patrones de entrada. El aprendizaje (no-supervisado) de auto-organización subyace en revelar las propiedades estadísticas de los patrones de entrada, creando así representaciones adecuadas para las características, por ejemplo vectores de peso, y automáticamente crear nuevos clusters. Los mapas neuronales compiten entre sí en búsqueda de ser activados al ganar los patrones de entrada. Solamente una neurona gana en cada iteración, convirtiéndose así en la ganadora de la BMU <sup>2</sup> [11].

Sea  $x_j$  el  $j$ -ésimo vector de características de entrada  $d$ -dimensional, y  $w_i$  el  $i$ -ésimo vector de peso  $d$ -dimensional. El primer paso del algoritmo es inicializar el vector de entrada usando un algoritmo de inicialización lineal. Los vectores de peso  $w_i$  definen la triangulación Voronoi del espacio de entrada

<sup>2</sup>Best Matching Unit, mejor unidad de emparejamiento

[1], [2]. Cada celda Voronoi está representada por su centroide, el cual corresponde al vector de pesos  $w_i$ . Cada patrón de entrada  $x_j$  está asignado a una celda Voronoi basada en la condición de vecino más cercano. Esto es, el índice BMU,  $c(j)$ , del patrón de entrada  $x_j$  definido en la Ecuación 9

$$c(j) = \arg \min(\|x_j - w_i\|) \quad (9)$$

donde  $\|$  denota distancia euclidiana. Asimismo, BL-SOM puede ser tratada como un método de cuantización de vector [12]. El paso más importante del algoritmo BL-SOM es la adaptación de los vectores de peso de las neuronas. Las neuronas están conectadas a neuronas adyacentes por una función de vecindario que indica la topología del mapa. Determina también, cuán fuertes son las conexiones entre las mismas [11]. En cada paso del entrenamiento, las neuronas actualizan la función de vecindario, cuyo propósito es correlacionar las direcciones de las actualizaciones de los pesos en un gran número de neuronas alrededor de la BMU [13].

Para actualizar las neuronas vencedoras y sus vecindarios, se puede optar ya sea por tipo de regla de adaptación LMS<sup>3</sup> o por un algoritmo *batch*. En el presente trabajo, se utiliza la última, la cual consiste en que un conjunto de entrenamiento dado  $x_j$ , el cual mantiene registro de las actualizaciones de los pesos, pero cuyo ajuste es aplicado solamente después de todas las muestras del conjunto que han sido consideradas. El aprendizaje cesa cuando se alcanza el número predeterminado de iteraciones [13]. En cada iteración del entrenamiento se determina la BMU. Tras ello, todas las neuronas que pertenecen al vecindario son actualizadas a su tiempo. La regla de actualización del  $i$ -ésimo vector de peso,  $w_i$ , es calculada mediante la Ecuación 10 [11]:

$$w_i(t+1) = \frac{\sum_{j=1}^N \alpha(t) h_{ic(j)}(t) x_j}{\sum_{j=1}^N h_{ic}(t)} \quad (10)$$

donde  $N$  define el número de patrones  $x_j$  que han sido asignados a la  $i$ -ésima neurona hasta la  $t$ -ésima iteración y  $h_{ic}(t)$  denota una función de vecindario alrededor del BMU ( $c(j)$ ). La tasa de aprendizaje  $a(t)$  es una función decreciente en el tiempo.

BL-SOM tiene algunas desventajas, tales como la falta de robustez en cuanto a valores atípicos y decisiones erróneas para el vector ganador debido a estimadores lineales. Pero esto se puede resolver utilizando estadísticas de orden multivariante [3].

### III-C. MMSOM

MMSOM calcula la mediana marginal de todos los patrones asignados a la neurona ganadora y actualiza solamente los vectores de peso BMU. MMSOM confía en el concepto de ordenamiento marginal. El ordenamiento marginal de los vectores de entrada  $N$   $x_1, x_2, \dots, x_n$  donde  $x_j = (x_{1j}, x_{2j}, \dots, x_{Dj})^T$

se realiza al ordenar los componentes de la neurona ganadora de forma independiente de acuerdo a las  $D$  dimensiones [3], [14]:

$$x_{q(1)} \leq x_{q(2)} \leq \dots \leq x_{q(N)}, q = 1, 2, \dots, D \quad (11)$$

donde  $q$  denota el índice del componente del vector en consideración. El nuevo vector de peso del BMU surge del cálculo de la mediana marginal de todos los patrones indexados en el BMU[15]. El cálculo de la mediana marginal está definido en la Ecuación (12)

$$\begin{aligned} \text{marginal\_median}\{x_1, x_2, \dots, x_n\} = \\ = \begin{cases} (x_{1(v+1)}, \dots, x_{D(v+a)})^T, & N = 2v + 1 \\ \left( \frac{x_{1(v)} + x_{1(v+a)}}{2}, \dots, \frac{x_{D(v)} + x_{D(v+a)}}{2} \right)^T, & N = 2v \end{cases} \end{aligned} \quad (12)$$

donde  $N$  denota el número de patrones asignados al BMU,  $w_c$ . La neurona ganadora es actualizada mediante la Ecuación 13:

$$w_c(t+1) = \text{marginal\_median}\{R_c(t) \cup x(t)\} \quad (13)$$

### III-D. VMSOM

VMSOM calcula el vector de medianas de todos los patrones asignados a la neurona ganadora y actualiza solamente el vector de peso del BMU. El operador de vector medio es el vector que pertenece al conjunto de vectores de entrada indexados por el BMU, el cual es el más cercano a todos los vectores de entrada actuales [16]. La mediana del vector de  $N$  vectores de entrada  $x_1, x_2, \dots, x_N$  está definida por la Ecuación 14:

$$x_l = \text{vector\_median}\{x_1, x_2, \dots, x_n\}$$

donde

$$l = \arg \min_k \sum_{j=1}^N |x_j - x_k| \quad (14)$$

La neurona ganadora es actualizada según la Ecuación 15:

$$w_c(t+1) = \text{vector\_median}\{R_c(t) \cup x(t)\} \quad (15)$$

## IV. OVMMMSOM

El modelo OVMMMSOM calcula un nuevo BMU para cada neurona, apartir de los BMU's resultantes de los entrenamientos por separado de MMSOM y VMSOM. El modelo se describe en el algoritmo 1.

## V. RESULTADOS EXPERIMENTALES

### V-A. Bases de Datos Utilizadas

Las pruebas han sido realizadas utilizando la base de datos de imagenes "COIL100"<sup>1</sup>, proporcionada por la Universidad de Columbia, la cual contiene 7200 imagenes correspondientes a 100 objetos, cada una de las imagenes muestra a un objeto rotado 72 veces, cinco grados a la vez como se muestra en las Figuras 1 y 2

<sup>1</sup>Columbia University Image Library  
http://www.cs.columbia.edu/CAVE/software/softlib/coil-100.php

<sup>3</sup>Least Mean Squared

**Algoritmo 1** Pseudocódigo del entrenamiento con OVMM-SOM

**Entrada:** dataset  $P_{inicial} = \{p_1, \dots, p_n\}$  no vacío

**Salida:** data clusterizada

- 1: **mientras** el valor de CDbw sea aceptable **hacer**
- 2:   Entrenar con MMSOM
- 3:   Obtener BMU:  $\overrightarrow{MMBMU}$
- 4:   Entrenar con VMSOM
- 5:   Obtener BMU:  $\overrightarrow{VMBMU}$
- 6:   Generar el nuevo BMU:  $\overrightarrow{OVMMBMU}$
- 7:    $\overrightarrow{OVMMBMU} = \lambda \overrightarrow{MMBMU} + (1 - \lambda) \overrightarrow{VMBMU}$
- 8:   Entrenar con OVMMSOM
- 9:   Calcular índice CDbw
- 10: **fin mientras**
- 11: **devolver** cierto



Figura 1. COIL100



Figura 2. Subconjunto de imágenes de COIL100 correspondiente a un objeto

**V-B. Pruebas**

La extracción de características de las imágenes se hizo utilizando el descriptor *ColorLayout* especificado en el Capítulo 2, el vector característica de cada imagen posee 12 dimensiones, 6 para luminancia y 6 para crominancia. El total de experimentos realizados para cada algoritmo fue de 100 entrenamientos. El entrenamiento de la red BL-SOM se hizo inicialmente con 1200 patrones (los últimos 12 de cada grupo de 72, cada grupo de 72 representa a un artículo del conjunto de COIL100), luego se agregaron los 6000 patrones restantes para clusterizar finalmente el conjunto total de patrones.

La validación de los clústers, para los 4 entrenamientos (BLSOM, MMSOM, VMSOM y la propuesta OVMM-SOM)

es realizada mediante el cálculo del índice CDbw.

La participación de cada uno de los entrenamientos (MM-SOM y VMSOM) en la propuesta esta definida por un índice  $\lambda$ , el cual definirá el nuevo BMU (OVMMBMU) de la siguiente manera:

$$\overrightarrow{OVMMBMU} = \lambda \overrightarrow{MMBMU} + (1 - \lambda) \overrightarrow{VMBMU} \quad (16)$$

Como se puede ver en las Figura 3 y 4, que representan los experimentos realizados (100 entrenamientos).

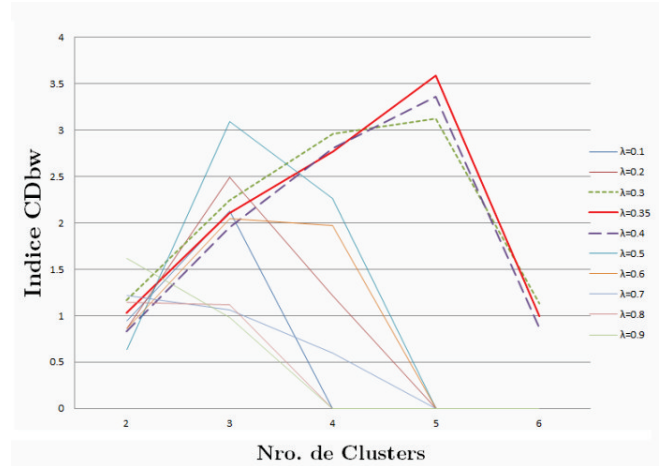


Figura 3. Promedio de los índices obtenidos con respecto al número de clusters utilizando OVMM-SOM

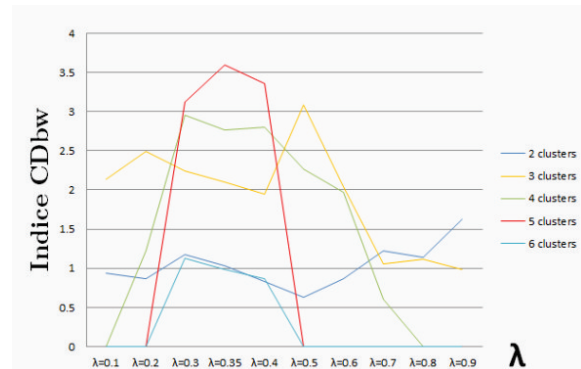


Figura 4. Comportamiento del índice CDbw para OVMM-SOM con respecto a los valores  $\lambda$

Tras los resultados mostrados en las Figuras 3 y 4, se elige 0,35 como valor del índice  $\lambda$  el cual se utilizará en las pruebas frente a los demás algoritmos. Los experimentos constan de entrenamientos con los 4 métodos: BL-SOM, MMSOM, VMSOM y OVMM-SOM. BL-SOM fue entrenada con 50 épocas cada vez. La Tabla I muestra el promedio de los resultados tras 100 experimentos.

El modelo OVMM-SOM propuesto para entrenamiento de SOM, con índice  $\lambda = 0,35$  ha demostrado mejorar el índice de



Tabla I  
RESULTADOS COMPARATIVOS USANDO LA DATA DE COIL100 (C =  
NÚMERO DE CLUSTERS OBTENIDOS)

Tam del mapa	BL		Métodos		SOM		OVM	
	CDbw	C	MM	C	VM	C	CDbw	C
4 × 4	1.76	2	<b>2.64</b>	2	1.54	3	2.38	3
5 × 5	1.93	4	<b>3.15</b>	3	2.89	4	2.48	5
6 × 6	2.07	5	2.61	3	2.77	4	<b>2.99</b>	5
7 × 7	2.38	4	<b>2.96</b>	5	2.56	4	2.65	4
8 × 8	2.59	5	2.49	4	3.12	5	<b>3.56</b>	5
9 × 9	2.29	5	3.42	4	3.28	4	<b>3.78</b>	5

validación de sus predecesores MMSOM y VMSOM, incluso superando el índice de validación que se le asignaba a los agrupamientos realizados con BL-SOM, por lo que los clústers obtenidos tras el entrenamiento propuesto por este modelo ofrecen grupos mas densos y aislados de los demás. Esto significa que, a diferencia de los demás métodos evaluados, ha proporcionado una detección más eficiente de clústers sobre la data, basado en el descriptor de color *ColorLayout*, como muestra la Figura 5.

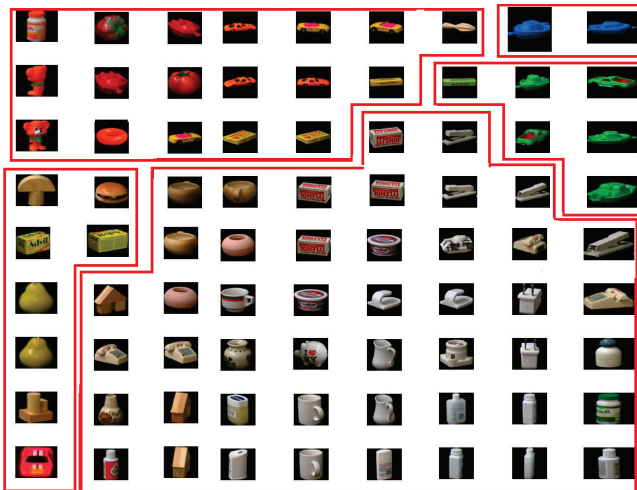


Figura 5. Clusterización de COIL100 con OVMMSOM y un mapa 9 × 9

## VI. CONSIDERACIONES

- En este artículo se presentó un modelo de agrupamiento basado en redes SOM aplicando los entrenamientos MMSOM y VMSOM junto al método de interpretación de la red: SL-SOM.
- Se probó la calidad de este nuevo modelo de agrupamientos en un conjunto de imágenes utilizando un descriptor de bajo nivel basado en color *Color Layout*.

## VII. CONCLUSIONES

- Se determinó experimentalmente que la unión de los entrenamientos MMSOM y VMSOM con un índice de participación  $\lambda$  llega a mejores resultados si es comparado con la aplicación de MMSOM y VMSOM por

separado, donde los resultados son el valor del índice de validación CDbw aplicado a los grupos resultantes en el conjunto de imágenes utilizado.

- Se encontró que para un valor de  $\lambda = 0,35$  se obtuvo el mejor rendimiento. Cabe destacar que en un inicio se propuso una participación por igual de los entrenamientos MMSOM y VMSOM pero los resultados con este escenario no fueron alentadores, por lo que se optó por definir el índice de participación  $\lambda$  y se buscó un valor que ofrezca mejores resultados.
- Otro punto a resaltar es la continuidad de OVMMSOM para obtener un número adecuado de clusters, pues en promedio ha encontrado la misma cantidad de clusters válidos sin depender del tamaño del mapa.

## REFERENCIAS

- [1] S. Haykin, "Neural networks: A comprehensive foundation," *Journal of Multi-Criteria Decision Analysis*, Upper Saddle River, Prentice-Hall, 1999.
- [2] T. Kohonen, "Self-organizing maps," *Springer-Verlag*, 2000.
- [3] I. Pitas, C. Kotropoulos, N. Nikolaidis, R. Yang, and M. Gabbouj, "Order statistics learning vector quantizer," *IEEE Trans. Image Processing*, vol. 5, pp. 1048-1053, 1996.
- [4] P. Salembier and J. R. Smith, "Mpeg-7 multimedia, description schemes," *Transcriptions on Circuits and Systems For video Technologies*, pags. 748-759., 2002.
- [5] N. Grira, M. Crucianu, and N. Boujemaa, "Unsupervised and semi-supervised clustering: a brief survey," *Review of Machine Learning Techniques for Processing Multimedia Content, Report of the MUSCLE European Network of Excellence*, 2004.
- [6] S. A. Elavarasi, J. Akilandeswari, and B. Sathiyabhama, "A survey on partition clustering algorithms," *International Journal of Enterprise and Bussiness Systems*, 01 2011.
- [7] S. Wu and T. Chow, "Clustering of the self-organizing map using a clustering validity index based on inter-cluster and intra-cluster density," *Pattern Recognition Journal*, pags. 175-188, 2003.
- [8] M. Halkidi and M. Vazirgiannis, "A density based cluster validity approach using multi-representatives," *Journal of Intelligent Information Systems*, 2008.
- [9] J. Costa, "Classificação automática e análise de dados por redes neurais auto-organizáveis," *PhD thesis, Universidade Estadual de Campinas.*, 1999.
- [10] A. Ultsch, "Self-organizing neural networks for visualization and classification," *Intell. Data Anal.* pp. 373-384, 1993.
- [11] Juha.Vesanto, J. Himberg, E. Alhoniemi, and J. Parhankangas, "Som toolbox for matlab 5," <http://www.cis.hut.fi/projects/somtoolbox>, Finland, 2000.
- [12] A. Gersho and R. Gray, *Vector Quantization and Signal Compression*, ser. Kluwer international series in engineering and computer science. Kluwer Academic Publishers, 1992. [Online]. Available: <http://books.google.com.pe/books?id=DwcDm6xgltUC>
- [13] M. M. V. Hulle, *Faithful Representations and Topographic Maps. From Distortion- to Information-Base Self-Organization*, 1st ed., ser. Adaptive and Learning Systems for Signal Processing, Communications and Control Series. New York: Wiley-Interscience, 2000.
- [14] C. Kotropoulos and I. Pitas, *Nonlinear Signal and Image Processing: Theory, Methods, and Applications*. Boca Raton, FL: CRC Press, 2004, ch. Self-organizing maps and their applications in image processing, information organization, and retrieval.
- [15] I. Pitas and P. Tsakalides, "Multivariate ordering in color image restoration," *IEEE Trans. Circuits and Systems for Video Technology*, pp. 247-259, 1991.
- [16] J. Astola, P. Haavisto, and Y. Neuvo, "Vector median filters," *Proceedings of the IEEE*, vol. 78, no. 4, pp. 678-689, 1990.