

Conjunto de datos de segmentación de imágenes

Andrés Felipe Montoya Bustamante, Juan Pablo Ospina Herrera

Departamento de Ingeniería de Sistemas

Universidad de Antioquia, Colombia

{andres.montoyab, juan.ospina3}@udea.edu.co

Resumen— En este proyecto se presenta todo el diseño, análisis y simulación de un sistema de predicción basado en técnicas de aprendizaje de máquina, describiendo el problema y su contexto en términos del estado del arte, especificando cada una de las etapas de desarrollo del trabajo, los modelos con sus respectivas restricciones, la metodología de validación utilizada, los resultados de todas las simulaciones y por último las conclusiones obtenidas luego de realizar todas las etapas del proyecto.

Palabras claves— Aprendizaje de máquina, segmentación de imágenes, simulación, modelo fenomenológico.

Abstract— This project presents the whole design, analysis and simulation of a prediction system based on machine learning techniques, describing the problem and its context in terms of the state of the art, specifying each stage of development of the work, the models with their respective restrictions, the validation methodology used, the results of all the simulations and finally the conclusions obtained after performing all the stages of the project..

Index Terms— Machine learning, Image segmentation, simulation, phenomenological model.

I. INTRODUCCIÓN

El aprendizaje de máquinas tiene como objetivo desarrollar técnicas que permitan a las computadoras *aprender* a partir de una información suministrada, facilitando a las personas la toma de decisiones. Actualmente encontramos diversos campos de aplicación para el aprendizaje de máquina, sin embargo esta se puede aplicar en cualquier área del conocimiento que requiera predicciones precisas para conocer el comportamiento de un sistema.

Un campo de aplicación muy común se encuentra en el análisis de imágenes a las cuales se les ha extraído un conjunto de características. En este

proyecto se busca entrenar un sistema que tome estas características y las clasifique en 7 clases diferentes según sea el caso, de tal manera que cuando lleguen nuevas instancias el sistema pueda predecir a qué clase pertenece cada muestra. Las 7 clases hacen referencia a 7 imágenes al aire libre desde las cuales se tomaron instancias de manera aleatoria, siendo cada instancia una región 3x3 de píxeles. Además, se debe destacar que cada instancia está descrita por 19 atributos numéricos de alto nivel.

Esta base de datos fue donada en 1990 al *Machine Learning Repository* por la universidad de Massachusetts, y se puede observar que es un problema de múltiples variables con 2310 instancias en el cual nuestra principal tarea es realizar una clasificación por clases de todas las muestras de entrenamiento para poder realizar predicciones de otras muestras que sean para validación.

Los atributos o variables con los que cuenta cada instancia son los siguientes:

1. region-centroid-col: La columna del píxel central de la región.
2. region-centroid-row: La fila del píxel central de la región.
3. region-pixel-count: El número de píxeles en una región (siempre será 9).
4. short-line-density-5: Los resultados de un algoritmo de extracto de línea que cuenta cuántas líneas de longitud 5 (en cualquier orientación) con bajo contraste, menor o igual a 5, pasan por la región.
5. short-line-density-2: Igual que el short-line-density-5, pero cuenta las líneas de alto contraste, mayores que 5.
6. vedge-mean: Promedio. Mide el contraste de los píxeles adyacentes horizontalmente en la región. Hay 6, se dan la media y la desviación estándar. Este atributo se utiliza como un detector de borde vertical.
7. vedge-sd: Desviación estándar. (Ver 6).
8. hedge-mean: Promedio. Mide el contraste de píxeles adyacentes verticalmente. Se utiliza para la detección de líneas

horizontales.

9. hedge-sd: Desviación estándar. (Ver 7).
10. intensity-mean: El promedio en la región de $(R + G + B) / 3$.
11. rawred-mean: El promedio sobre la región del valor de R.
12. rawblue-mean: El promedio sobre la región del valor de B.
13. rawgreen-mean: El promedio sobre la región del valor de G.
14. exred-mean: Mide el exceso de rojo: $(2R - (G + B))$.
15. exblue-mean: Mide el exceso de azul: $(2B - (G + R))$.
16. exgreen-mean: Mide el exceso de verde: $(2G - (R + B))$.
17. value-mean: Valor promedio. 3-d transformación no lineal de RGB. (El algoritmo se puede encontrar en Foley y VanDam, fundamentos de gráficos de computadora interactivos).
18. saturatoin-mean: Saturación promedio. (Ver 17).
19. hue-mean: Matiz promedio. (Ver 17).

Cabe destacar que todas las variables están medidas en números reales.

Por último, solo se cuenta con una variable a predecir la cual corresponde a la imagen a la cual corresponde cada muestra. Esta variable solo puede tomar 7 valores los cuales son: Ladrillo, cielo, follaje, cemento, ventana, camino, hierba (brickface, sky, foliage, cement, window, path, grass). Se puede observar que las 7 clases están dadas en texto, por lo tanto se hizo necesario realizar la siguiente codificación:

- | | |
|-------------|---|
| • Brickface | 1 |
| • Sky | 2 |
| • Foliage | 3 |
| • Cement | 4 |
| • Window | 5 |
| • Path | 6 |
| • Grass | 7 |

Cabe destacar que en este caso no tiene sentido las distancias entre el valor de cada clase, por lo tanto se asignó estos valores de manera aleatoria. Además, se resalta que para el algoritmo basado en redes neuronales esta codificación no sirve, por lo tanto se utiliza one_hot_encoding quedando cada variable del siguiente modo: Brickface=1000000, Sky=0100000, y así sucesivamente hasta llegar a Grass= 0000001.

Finalmente se debe aclarar que la base de datos no cuenta con ningún valor faltante.

II. ESTADO DEL ARTE

Luego de realizar una búsqueda exhaustiva en bases de datos bibliográficas como sciencedirect, ieeexplore, acm digital y google scholar se encontró que existen diversos artículos que trabajan problemas de predicción similares al que se trabaja en este proyecto. Para este proyecto de investigación se encontró seis artículos que realizan un proceso de clasificación sobre datos basados en reconocimiento de imágenes, entre los cuales dos usan la misma base de datos escogida por nosotros, y los otros cuatro utilizan bases de datos diferentes pero tratan problemas similares.

Inicialmente se encontró un artículo publicado en Science Direct llamado Fast Hierarchical clustering and its validation en cual se implementaron los métodos Histograma y Slope, con los cuales se obtuvo una eficiencia de 99.5% y 98.6% respectivamente.

Además, se encontró un artículo publicado en el International Journal of Current Research en el cual realizan varios experimentos sobre la misma base de datos que estamos trabajando y usan cinco clasificadores basados en árboles de decisión (NB Tree, BF Tree, J48, FT tree y J48graft) con el objetivo de comparar los resultados obtenidos en cada uno. En cada modelo utilizan validación cruzada y encuentran la matriz de confusión para poder obtener la eficiencia. En este caso se obtuvo como mejor clasificador el J48graft con un 96.9697% de eficiencia en prueba.

También se halló un artículo publicado por el departamento de ciencias de la computación e ingeniería de la información de la Universidad HungKuang, Taichung, Taiwán; en el cual el principal objetivo es hacer uso de las redes neuronales de decisión basadas en el aprendizaje de múltiples instancias para la recuperación y clasificación de imágenes^[1], utilizando cross-validation con 10 folds como metodología de validación y usando como su principal medida de desempeño el área bajo la curva de la conocida gráfica ROC, se obtuvo como valor máximo 93.87% de área bajo la curva.

Por otro lado también se encontró el artículo publicado por la Universidad de Túnez - El Manar, ubicada en la capital de Túnez, en la cual utilizaron una red neuronal artificial (ANN) para resolver el problema clasificación de la densidad mamaria mamográfica utilizando un paradigma clasificador combinado^[2], usando como metodología de

validación cross-validation con 10 folds y empleando como medida la tasa de reconocimiento generado por la red neuronal, como resultado se pudo obtener que la mayor tasa de reconocimiento fue del 96.7%.

Posteriormente se encontró un artículo en el cual se busca utilizar máquinas de soporte de vectorial para la clasificación de imágenes basadas en histograma^[3], este fue publicado por Olivier Chapelle, y en el cual se aprecia que es diferente a los anteriores debido a que este usa máquinas de soporte vectorial como técnica de aprendizaje, en vez de redes neuronales y su principal medida es la tasa de error mínima, en este artículo se alcanzaron la tasa mínima de error del 11%.

Por último, se analizó el artículo publicado por la escuela de medio ambiente y sostenibilidad de la universidad de Saskatchewan, en Canadá, en el cual se busca la comparación de análisis de imágenes basadas en píxeles y objetos con algoritmos de aprendizaje de máquina seleccionados para la clasificación de paisajes agrícolas utilizando imágenes SPG-5 HRG^[4], haciendo uso de varias técnicas de aprendizaje para posteriormente comparar los resultados, además este utiliza cross-validation con 10 folds como metodología de validación y usando dos medidas de desempeño: precisión y el coeficiente Kappa. Para el algoritmo basado en árboles de decisión se logró para modelos basados en píxeles y basados en objetos 85,4% y 83,3%, respectivamente como máximo valores de precisión. Para el caso del algoritmo random forest se obtuvo como máximo valor de precisión para modelos basados en píxeles 91.1% y para modelos basados en objetos un 93.1%. Y para el caso del algoritmo de máquinas de soporte vectorial se obtuvo el siguiente porcentaje de precisión: 89.8% y 91.4%, para modelos basados en píxeles y objetos respectivamente.

Se puede observar que todas las medidas de desempeño evaluadas en los 4 artículos presentan muy buenos resultados, obteniendo altos porcentajes de precisión y eficiencia y bajos porcentajes de error en cada caso.

III. EXPERIMENTOS

Ahora se procede a explicar ciertos aspectos importantes que se deben tener en cuenta antes de evaluar cualquier tipo de solución. Como se describió en la introducción, la base de datos^[5] que se utilizó para la realización de experimentos fue donada en 1990 al Machine Learning Repository

por la universidad de Massachusetts, y se puede observar que es un problema de múltiples variables con 2310 muestras. Cuenta con 19 variables de entrada y una única variable de salida. Sin embargo se pudo apreciar al analizar la base de datos que la característica 3 (region-pixel-count) es una constante ya que siempre toma el valor de 9, por lo tanto se tomó la decisión de eliminarla ya que no aporta ninguna información para la toma de decisiones, por tanto se trabaja con 18 variables de entrada. Este problema es de clasificación, y consta de 7 clases con 330 muestras por cada clase. Se puede apreciar que la base de datos está balanceada por lo tanto no es necesario implementar técnicas de sobremuestreo y submuestreo inteligente. Por otra parte se definió como metodología de validación cross-validation con 10 folds.

En este proyecto se evaluó el uso de 5 modelos de predicción, con los cuales se obtuvo cada matriz de confusión para así obtener la eficiencia y los intervalos de confianza.

Para el algoritmo de funciones discriminantes gaussianas se utilizó la función que trae por defecto matlab que se llama classify. Haciendo uso de dicha función se obtuvo una eficiencia en prueba del 0.94084. Además, se implementó el modelo de mezclas gaussianas en el cual se obtuvo los siguientes resultados:

Matriz de covarianza	Mezclas	Eficiencia
Completa	1	0.66407 +- 0.036744
	2	0.68528 +- 0.13853
	3	0.76494 +- 0.10533
	4	0.7974 +- 0.10072
	5	0.82035 +- 0.086922
Diagonal	1	0.79827 +- 0.027996
	2	0.8697 +- 0.024527
	3	0.85152 +- 0.020104
	4	0.88355 +- 0.029036
	5	0.90216 +- 0.027847
	6	0.8987 +- 0.044392
Esférica	1	0.7632 +- 0.032845
	2	0.80823 +- 0.019791
	3	0.8342 +- 0.024914
	4	0.83939 +- 0.034779

	5	0.85065 +- 0.028295
	6	0.85671 +- 0.023308

La matriz de confusión obtenida con una matriz de covarianza diagonal y 5 mezclas fue:

33	0	0	0	0	0	0
0	31	0	0	0	0	0
0	0	15	1	2	0	0
0	1	1	33	1	0	0
1	0	8	0	33	0	0
0	0	0	0	0	36	0
0	0	0	0	0	0	35

Utilizando el algoritmo de los K vecinos más cercanos, se evaluó variando el valor de K. Los resultados fueron los siguientes:

K	Eficiencia
1	0.96667 +- 0.014437
2	0.95801 +- 0.015414
3	0.95584 +- 0.012038
4	0.95238 +- 0.017555
5	0.94286 +- 0.02109
6	0.94069 +- 0.018597
7	0.9381 +- 0.020514

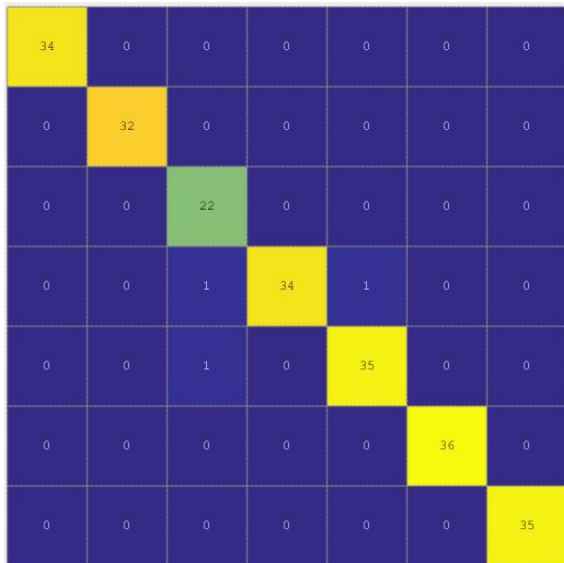
Y la matriz de confusión obtenida con k igual a 4 fue:

33	0	0	0	1	0	0
0	32	0	0	0	0	0
0	0	22	0	2	0	0
0	0	1	31	0	0	0
1	0	1	1	33	0	0
0	0	0	2	0	36	0
0	0	0	0	0	0	35

Evaluando el algoritmo random forest, y variando el número de árboles, se obtuvieron estos resultados:

Número de árboles	Eficiencia
10	0.97446 +- 0.0069053
20	0.97576 +- 0.0082137
30	0.97749 +- 0.0083644
40	0.97749 +- 0.0067065
50	0.97792 +- 0.0080213
100	0.98009 +- 0.0091719
500	0.98009 +- 0.00769

La matriz de confusión obtenida con 100 árboles fue:

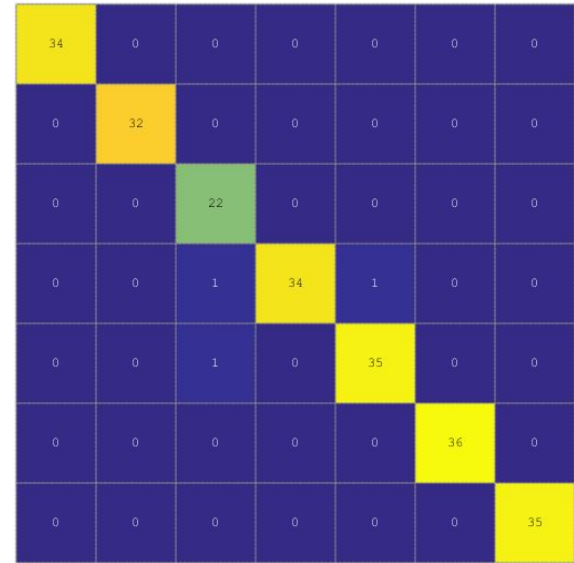


Haciendo uso del algoritmo de redes neuronales artificiales, utilizando solo una capa oculta y variando los parámetros número de neuronas por capa y número de épocas, se obtuvo el siguiente resultado:

Épocas	Neuronas	Eficiencia
100	34	0.9671 +- 0.010244
	35	0.9658 +- 0.011803
	36	0.97056 +- 0.012874
	37	0.97013 +- 0.015058
	38	0.9684 +- 0.015548

Cabe resaltar que no fue necesario variar el número de épocas ya que el algoritmo converge antes de alcanzar las 100 épocas. También se debe recordar el problema cuenta con 18 entradas, por lo cual se utilizó alrededor de 36 neuronas en la capa oculta.

La matriz de confusión obtenida con 36 neuronas es:

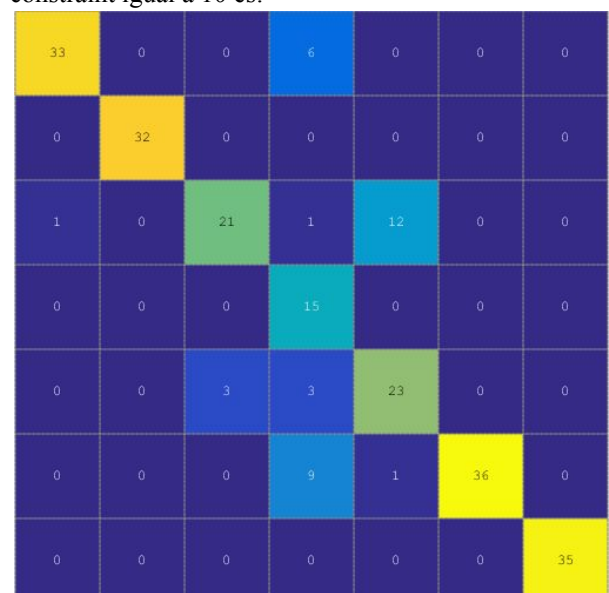


Ahora utilizaremos el algoritmo basado en máquinas de soporte vectorial. En esta parte se analiza con diferentes tipos de kernel (lineal y gaussiano) y se varía el box constraint y el gamma (según sea el caso).

Kernel lineal

Box constraint	Eficiencia
0.01	0.8316 +- 0.020248
0.1	0.83074 +- 0.02135
1	0.8381 +- 0.021998
10	0.84416 +- 0.022724
100	0.84719 +- 0.021889

La matriz de confusión obtenida con un box constraint igual a 10 es:



Kernel gaussiano

Box constraint	Gamma	Eficiencia
0.01	0.01	0.35714 +- 0.058624
	0.1	0.81342 +- 0.039331
	1	0.87446 +- 0.025813
	10	0.84545 +- 0.027383
	100	0.76883 +- 0.029304
0.1	0.01	0.46797 +- 0.057628
	0.1	0.87532 +- 0.019553
	1	0.92814 +- 0.022743
	10	0.89481 +- 0.019896
	100	0.83636 +- 0.022149
1	0.01	0.53939 +- 0.061736
	0.1	0.8961 +- 0.015675
	1	0.96061 +- 0.016884
	10	0.9329 +- 0.015575
	100	0.88961 +- 0.0232
10	0.01	0.55801 +- 0.043408
	0.1	0.90303 +- 0.017098
	1	0.971 +- 0.012752
	10	0.95455 +- 0.016982
	100	0.90779 +- 0.01708
100	0.01	0.55801 +- 0.044263
	0.1	0.90606 +- 0.016075
	1	0.97013 +- 0.013296
	10	0.96753 +- 0.012455
	100	0.92857 +- 0.016101

La matriz de confusión obtenida con un box constraint igual a 10 y gamma igual a 1 es:

34	0	0	0	0	0	0
0	30	0	0	0	0	0
0	2	23	0	0	0	0
0	0	0	33	0	0	0
0	0	1	0	36	0	0
0	0	0	1	0	36	0
0	0	0	0	0	0	35

IV. ANÁLISIS INDIVIDUAL

En esta sección se procede un análisis individual de cada una de las características, a partir de medidas de correlación y del índice de Fisher para identificar cuáles variables son candidatas a ser eliminadas.

A. Análisis por medio de medidas de correlación de Pearson.

Una vez ejecutado el código de medidas de correlación de Pearson sobre la base de datos a usar, se obtuvieron las siguientes correlaciones entre las características:

La característica número 10 explica a las características 11, 12, 13 y 17. La característica número 11 explica a la 12, 13 y 17. La característica número 12 explica a la 13 y a la 17. Por último la características 13 explica a la 17.

Con base en lo anterior y según la correlación entre las características se decidieron como candidatas a eliminar las variables 11, 12, 13 y 17 ya que estas tienen una correlación evidente y están descritas por otras características.

B. Análisis por medio de cociente discriminante de fisher.

El cociente discriminante de fisher muestra la capacidad discriminativa que tiene cada una de las características que pertenecen a la base de datos. Aquellas características con una capacidad discriminativa aceptable serán aquellas que tengan un índice mayor a 0.5.

A continuación se mostraran los respectivos cocientes de fisher para cada una de las características:

Característica	Cociente discriminante de Fisher
1	0,66361
2	1
3	0,38521
4	0,38926
5	0,32862
6	0,12368
7	0,32059
8	0,091045
9	0,22149
10	0,21824
11	0,22835
12	0,21603
13	0,51193
14	0,31128
15	0,39549
16	0,24127
17	0,45076
18	0,38939

En base a los resultados anteriores se observó que las características con una capacidad discriminativa aceptable fueron las número 1, 2 y 13 por lo cual estas características permanecerán en el modelo y el resto de características son candidatas a ser eliminadas.

Por último debemos observar que en ambos análisis se obtuvieron resultados diferentes, sin embargo en ambos se concluye que las características 11, 12 y 17 son candidatas a eliminarse.

V. SELECCIÓN DE CARACTERÍSTICAS

En esta sección se abordará el tema de la selección de características, para esto usamos un metodo de busqueda secuencial ascendente con una función criterio tipo wrapper .

Usaremos wrapper debido a que usualmente se alcanzan mejores tasas de predicción y además tienen una mayor capacidad de generalización que los criterio tipo filtro.

En la función tipo wrapper utilizaremos un modelo random forest con 100 árboles, debido a que este modelo fue el que presentó la mejor eficiencia en la etapa de evaluación de modelos.

Se observó que al ejecutar el método de búsqueda secuencial ascendente las variables seleccionadas eran diferentes entre cada repetición, por lo cual se realizó diez repeticiones a este metodo de busqueda y aquellas que aparecieron cero o una vez, serán características candidatas a eliminar.

Característica	Número de apariciones
1	10
2	10
3	5
4	4
5	0
6	0
7	0
8	0
9	1
10	4
11	4
12	3
13	0
14	2
15	2
16	2
17	9
18	10

Con base en la anterior se observa que las características candidatas a ser eliminadas del conjunto de datos de entrada serán las variables número 5, 6, 7, 8, 9 y 13. Con esto se puede concluir que el porcentaje de reducción fue del 33.33%, lo que quiere decir que ahora se cuenta con un 66.66% del total de características iniciales.

Ahora se procede a evaluar los modelos que obtuvieron mejor eficiencia con el conjunto de variables resultantes luego de la aplicación del método de búsqueda secuencial.

ID	Eficiencia
----	------------

1	0.98225 +- 0.0094514
2	0.97273 +- 0.015563
3	0.96883 +- 0.013351

Donde el ID hace referencia a:

1. Random Forest con 100 árboles
2. RNA con 36 Neuronas y 100 épocas
3. SVM con box constraint 10 y gamma 1

Esto debido a que fueron los 3 mejores modelos encontrados en la etapa de evaluación.

Por último podemos apreciar que la eficiencia aumentó cuando se evaluó los modelos Random Forest y redes neuronales artificiales, y por el contrario en el caso de máquinas de soporte vectorial disminuyó. Sin embargo las diferencias fueron muy mínimas en los tres casos.

VI. EXTRACCIÓN DE CARACTERÍSTICAS

En esta sección se utilizará el método de PCA para realizar extracción de características. Para definir los componentes principales se evaluarán los 3 mejores modelos con un 85%, 90% y 95% de varianza acumulada. Esto debido a que se busca mantener una alta variabilidad respecto a la información original. En este caso se vuelve a evaluar los 3 modelos de la sección anterior.

Porcentaje de varianza	ID	Eficiencia
85%	1	0.82121 +- 0.078057
	2	0.80102 +- 0.05234
	3	0.68442 +- 0.10403
90%	1	0.93117 +- 0.019084
	2	0.91023 +- 0.01623
	3	0.86277 +- 0.026845
95%	1	0.93117 +- 0.019084
	2	0.91023 +- 0.01623
	3	0.86277 +- 0.026845

Apesar de que la eficiencia en todos los modelos evaluados fue menor, se obtuvo un porcentaje de reducción del 77.7% cuando el umbral del porcentaje de varianza es del 85%, y para un

porcentaje de varianza del 90% y el 95% se obtuvo una reducción del 72.2%, lo cual representa una reducción bastante alta con respecto al número de características iniciales, lo que indica que el conjunto de variables original cuenta con alta dependencia.

VII. DISCUSIÓN

Luego de analizar los resultados obtenidos en la sección III, podemos concluir que los mejores modelos son: Random Forest con 100 árboles que obtuvo 98.009% de eficiencia, luego tenemos redes neuronales artificiales con 36 neuronas y 100 épocas que obtuvo una eficiencia en prueba de 97.056% y por último tenemos las máquinas de soporte vectorial con box constraint 10 y gamma 1 que obtuvo un 97.1% de eficiencia. Se puede apreciar que en estos 3 modelos se obtuvo un porcentaje de eficiencia bastante alto, y sin duda el Random Forest con 100 árboles fue el que tuvo mayor eficiencia en prueba (98.009%).

Aplicando la técnica de selección de características sobre nuestros tres mejores modelos pudimos observar que para el modelo de Random Forest con 100 árboles la eficiencia incrementó aproximadamente en un 0.216%, para el modelo de Redes Neuronales Artificiales con 36 neuronas se produjo un incremento del 0.217% en su eficiencia y para el modelo de máquinas de soporte vectorial con box constraint 10 y gamma 1 la eficiencia disminuyó en un 0.217%. Por lo tanto, se decide aplicar esta técnica debido al aumento en la eficiencia que trae para los dos primeros modelos.

Luego de aplicar la técnica de extracción de características (PCA) se pudo notar que la eficiencia en prueba disminuyó en todos los casos, a pesar de que fue en pequeña escala. Sin embargo se redujo en entre un 70% y 80% las características originales, lo cual es una reducción notable. Si comparamos con el artículo *A Greedy EM Algorithm for Gaussian Mixture Learning* en el cual utilizan la misma base de datos que nosotros, ellos llegaron a que con una reducción del 72.2% se explica el 95% de los datos originales, lo cual coincide con nuestro resultado. A pesar de esto, esta técnica no se decide aplicar debido a la reducción en la eficiencia.

Ahora se procede a comparar los resultados obtenidos en la sección II en la cual se describió el estado del arte. Para esta comparación solo se incluyó los resultados obtenidos de los artículos que trabajan sobre la misma base de datos que la nuestra. Además se incluye el modelo que mejor resultado tuvo durante toda la realización del

proyecto.

Modelo	Eficiencia
Histograma	99.5%
Slope	98.6%
J48graft	96.9697%
Random Forest 100 árboles	98.225%

A partir de los resultados obtenidos en la anterior tabla, podemos concluir que los modelos utilizados en *Fast Hierarchical clustering and its validation* obtuvieron mayor porcentaje de eficiencia, siendo el modelo basado en histogramas el mejor con un 99.5%. Se debe destacar que las predicciones brindadas con un modelo Random Forest con 100 árboles son bastantes altas y se pueden utilizar en la toma de decisiones.

REFERENCIAS

- [1] Manoranjan Dash, Huan Liu, Peter Scheuermann, Kian Lee Tan, ScienceDirect Data & Knowledge Engineering Vol 44, January 2003, Fast Hierarchical clustering and its validation
- [2] Nikos Vlassis, Aristidis Likas, Neuronal Processing Letters 2002, A Greedy EM Algorithm for Gaussian Mixture Learning.
- [3] Dr. P. K. Srimani, F. N. Sc, Shanthi Mahesh. International Journal of Current Research Vol. 4, Issue, 09, pp.135-140, September, 2012. KNOWLEDGE DISCOVERY IN IMAGE-SEGMENTATION DATA SET USING DECISION TREE CLASSIFIERS.
- [4] Yeong-Yuh Xu, Neurocomputing, Multiple-instance learning based decision neural networks for image retrieval and classification.
- [5] Keir Bovis and Sameer Singh, PANN Research, Classification of Mammographic Breast Density Using a Combined Classifier Paradigm.
- [6] Olivier Chapelle, Patrick Haffner, and Vladimir N. Vapnik, IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 10, NO. 5, SEPTEMBER 1999, Support Vector Machines for Histogram-Based Image Classification.
- [7] Dennis C. Duro, Steven E. Franklin, Remote Sensing of Environment, A comparison of pixel-based and object-based image analysis with selected machine learning algorithms for the classification of agricultural landscapes using SPOT-5 HRG imagery.
- [8] UCI Machine Learning Repository, Image Segmentation Data Set. Tomado el 20 de abril de: <https://archive.ics.uci.edu/ml/datasets/Image+Segmentation>.