

Clasificación automática de las vocales en el lenguaje de señas colombiano

Automatic classification of vowels in Colombian sign language

Deivid J. Botina-Monsalve¹, María A. Domínguez-Vásquez², Carlos A. Madrigal-González³ y Andrés E. Castro-Ospina⁴

Recibido: 10 de octubre de 2017
Aceptado: 13 de diciembre de 2017

Cómo citar / How to cite

D. J. Botina-Monsalve, M. A. Domínguez-Vásquez, C. A. Madrigal-González y A. E. Castro-Ospina, Clasificación automática de las vocales en el lenguaje colombiano de señas. *TecnoLógicas*, vol. 21, no. 41, pp. 103-114, 2018.

¹ Ingeniero Electrónico, Facultad de Ingenierías, Departamento de Electrónica y Telecomunicaciones, Instituto Tecnológico Metropolitano – ITM, Medellín, deividbotina@itm.edu.co

² Ingeniera Electrónica, Facultad de Ingenierías, Departamento de Electrónica y Telecomunicaciones, Instituto Tecnológico Metropolitano – ITM, Medellín-Colombia, mariadominguez@itm.edu.co

³ PhD en Ingeniería de Sistemas, Ingeniero Electrónico, Facultad de Ingenierías, Departamento de Electrónica y Telecomunicaciones, Instituto Tecnológico Metropolitano – ITM, Medellín-Colombia., carlosmadrigal@itm.edu.co

⁴ MSc en Automatización Industrial, Ingeniero Electrónico, Grupo de Investigación Automática, Electrónica y Ciencias Computacionales, Instituto Tecnológico Metropolitano – ITM, Medellín-Colombia, andrescastro@itm.edu.co



Resumen

El reconocimiento del lenguaje de señas es un problema de alta complejidad, debido a la cantidad de gestos estáticos y dinámicos necesarios para representar dicho lenguaje, teniendo en cuenta que el mismo variará para cada país en particular. Este artículo se enfoca en el reconocimiento de las vocales del lenguaje colombiano de señas, de forma estática. Se adquirieron 151 imágenes por cada clase, teniendo en cuenta también una clase no vocal adicional con diferentes escenas. A partir de cada imagen capturada se separa el objeto de interés del resto de la escena usando información de color; luego, se extraen características para describir el gesto correspondiente a cada vocal o a la clase que no corresponde a ninguna vocal. Posteriormente, se seleccionan cuatro conjuntos de características. El primero con la totalidad de ellas; a partir de este salen tres nuevos conjuntos: el segundo extrayendo un subconjunto de características mediante el algoritmo de Análisis de Componentes Principales (PCA). El tercer conjunto, aplicando Selección Secuencial hacia Adelante (SFS), mediante la medida de FISHER y el último conjunto con SFS basado en el desempeño del clasificador de los vecinos más cercanos (KNN). Finalmente se prueban múltiples clasificadores para cada conjunto por medio de validación cruzada, obteniendo un desempeño superior al 90% para la mayoría de los clasificadores, concluyendo que la metodología propuesta permite una adecuada separación de las clases.

Palabras clave

Análisis de componentes principales, clasificación, lenguaje de señas colombiano, selección de características, validación cruzada.

Abstract

Sign language recognition is a highly-complex problem due to the amount of static and dynamic gestures needed to represent such language, especially when it changes from country to country. This article focuses on static recognition of vowels in Colombian Sign Language. A total of 151 images were acquired for each class, and an additional non-vowel class with different scenes was also considered. The object of interest was cut out of the rest of the scene in the captured image by using color information. Subsequently, features were extracted to describe the gesture that corresponds to a vowel or to the class that does not match any vowel. Next, four sets of features were selected. The first one contained all of them; from it, three new sets were generated. The second one was extracted from a subset of features given by the Principal Component Analysis (PCA) algorithm. The third set was obtained by Sequential Feature Selection (SFS) with the FISHER measure. The last set was completed with SFS based on the performance of the K-Nearest Neighbor (KNN) algorithm. Finally, multiple classifiers were tested on each set by cross-validation. Most of the classifiers achieved a performance over 90%, which led to conclude that the proposed method allows an appropriate class distinction.

Keywords

Principal Component Analysis, Classification, Colombian sign language, Feature selection, Cross-validation.

1. INTRODUCCIÓN

El lenguaje de señas es importante para establecer la comunicación entre personas con discapacidad auditiva. Según el censo realizado en Colombia en el 2015, la comunidad sorda colombiana está compuesta por aproximadamente 455.718 personas [1], las cuales, debido a su condición, poseen un lenguaje propio donde se expresan mediante gestos de manos y cara, y su lenguaje corporal. Gran parte de esta comunicación se realiza mediante el movimiento, posición determinada de las manos y ocasionalmente con expresiones faciales, esto es conocido como el lenguaje de señas. Este lenguaje, contrario a lo que se piensa, no es universal. Muy pocos habitantes de Colombia comprenden el lenguaje colombiano de señas [2], lo que dificulta la comunicación entre la comunidad sorda colombiana con el resto de personas, lo cual conlleva a que se aislen o que interactúen en la mayoría del tiempo solo con personas que presentan su misma condición. Además, para las personas con discapacidad auditiva entender el español escrito les representa aprender una segunda lengua, la cual posee una distinción bastante notable, como es la expresión oral en vez de gestual y la recepción auditiva frente a la visual.

Debido a que el lenguaje de señas es particular para cada país, la comunidad de reconocimiento de patrones ha trabajado sobre el problema de reconocimiento automático de señas en varios países, sin embargo, aún existen algunos retos como el desarrollo de descriptores espacio-temporales robustos y simples que permitan hacer el reconocimiento de forma fiable y en tiempo real.

En este trabajo se desarrolló una metodología para la clasificación de las vocales en el lenguaje colombiano de señas. Dentro de los trabajos encontrados en la literatura, se encuentra que múltiples autores han trabajado en la clasificación de gestos. Smith et al. [3] propusieron una patente

para detectar y reconocer los gestos de la mano sobre una imagen a partir de una serie de pasos: se toma la información de la imagen, se extrae el color de la piel en el espacio de color RGB para detectar el movimiento, de esta manera al final es posible reconocer los gestos dentro de la imagen. De manera similar, en [4], Fang et al, realizaron reconocimiento de los gestos de la mano, pero en tiempo real.

Premaratne et al. [5] proponen el uso de “primitivos gestuales” para la clasificación de posturas de la mano, utilizando las librerías de OpenCV. Hruz et al. [6] utilizan Patrones Binarios Locales (LBP) para demostrar su eficacia al clasificar la forma de la mano. Sin embargo, aunque encontrar de forma correcta la mano y reconocer gestos es un primer enfoque válido, cabe notar que se reconocen gestos, los cuales pueden ser de cualquier tipo o naturaleza, no un conjunto finito, como lo es el alfabeto de lenguaje de señas.

Huang et al. usaron métodos de aprendizaje profundo o Deep Learning, como son las redes neuronales convolucionales [7], al igual que Pigou et al. en [8], para el reconocimiento de algunas palabras en el lenguaje de señas, en donde los datos son adquiridos por medio de un sensor Kinect el cual proporciona información del espacio tridimensional. Basados en el uso de este sensor, se usan convoluciones 3D favorables para realizar un apareo entre la trayectoria de la palabra gesticulada en el lenguaje de señas con su respectiva etiqueta. También, Rao et al. realizaron un reconocimiento de señas de la India con redes neuronales en tiempo real, utilizando un smartphone [9].

En [10], se propuso un software interactivo con el cual pueden traducir lo que una persona muda exprese mediante señas del lenguaje chino. Para el reconocimiento de las señas se utiliza el sensor Kinect donde el resultado es visualizado en forma de letras chinas mediante una pantalla. Siendo el caso contrario cuando se desea traducir de letras a señas, el mensaje se ingresa-

rá por teclado para ser interpretado por el avatar del software. Los métodos descritos requieren que solo esté en movimiento una extremidad, por lo que Lim et al. [11] introducen un método que se enfoca en aislar ambas extremidades por medio del Filtro Serial de Partículas (SPF), para realizar el reconocimiento del lenguaje de señas.

En [12] se hace una recopilación de los aspectos clave para el reconocimiento del lenguaje de señas con seguimiento y sin seguimiento de los movimientos; tomando en cuenta no solo características de las manos, sino también expresiones faciales y corporales. Para la clasificación se utilizan redes neuronales además de modelos ocultos de Márkov (HMM), el cual requiere grandes conjuntos de datos para su entrenamiento; una de sus principales aplicaciones de HMM son el reconocimiento del habla, gestos y movimientos corporales. Por su parte, en el trabajo presentado en [13], usan árboles de patrones secuenciales (SP-Tree) para la clasificación del lenguaje de señas, dicha técnica es eficiente para el aprendizaje de conjuntos SP-Tree representando patrones espacio-temporales de manera eficaz, para abordar problemas de clasificación multiclase por medio del ensamble de clasificadores débiles o simples; también realizan un comparativo con los HMM, usado habitualmente en el reconocimiento de señas.

Agarwal et al. [14] realizan un sistema para el reconocimiento de los números chinos en el lenguaje de señas, el cual hace uso de las características de profundidad y velocidad adquiridas por medio de un sensor Kinect. La clasificación de las señas la hacen por medio de Máquinas de Vectores de Soporte (SVM) multi-clase, también desarrollan un comparativo entre la exactitud de la clasificación y el tiempo de entrenamiento para diferentes subconjuntos de los gestos. En otro trabajo, proponen un método para el reconocimiento del lenguaje de señas del alfabeto árabe basado en un controlador de movimiento de salto (LMC) el cual detecta el movimiento de la mano y

los dedos, entregando información de la posición y velocidad, este evita que la adquisición del gesto de la mano se vea perturbado por cambios en el ambiente; luego de la adquisición y extracción de características implementan dos clasificadores y realizan un comparativo entre redes neuronales perceptrón multicapa (MLP) y el clasificador bayesiano simple (NBC) [15].

Jeroen F. et al. [16] clasifican el lenguaje de señas dinámico holandés, teniendo en cuenta expresiones faciales y corporales, donde el número total de características puede ser de hasta 1500 por seña. Los métodos implementados para la clasificación de las 120 clases fueron selección de características discriminantes (DF), combinación de detectores de características discriminantes (CDFD) y clasificación cuadrática de características discriminantes en el mapeo Fisher (Q-DFFM), implementando dos métodos probabilísticos, estadística dinámica de alineamiento temporal (SDTW) y el modelo oculto de Márkov realizan combinaciones múltiples de los modelos de salida de las clases, siendo esta una práctica común para incrementar el desempeño. En otras aplicaciones de clasificación automática, se han utilizado métodos de experimentación similares [17] [18] [19].

Los métodos descritos requieren de algoritmos robustos por la cantidad de datos manejados, y a medida que se extienden las bases de datos se hace necesario su implementación en una unidad de procesamiento gráfico (GPU) para acelerar el cálculo computacional.

Con el fin de crear herramientas tecnológicas para el reconocimiento de vocales en el lenguaje colombiano de señas, se propone identificar dichas vocales en un fondo uniforme por medio de varios clasificadores. A partir de una base de datos de 151 ejemplos para cada una de las 6 clases (5 vocales y una clase adicional de no vocal), se toman cuatro conjuntos de características, el *Conjunto 1* compuesto por 12 características de forma, obtenidas por las

librerías de CVBlobslib.lib de OpenCV (perímetro, centro en Y, amplitud, compacidad, diferencia en X, diferencia en Y, excentricidad, área, longitud eje menor, longitud, longitud eje mayor y momento), más 54 características adicionales extraídas por el método propuesto por Pradeep et al. [20]. El *Conjunto 2* consta de 28 características extraídas que son el resultado de aplicar el algoritmo de análisis de componentes principales (PCA) [21] al primer conjunto. El *Conjunto 3* se obtiene al aplicar al *Conjunto 1* una selección de características mediante Sequential Forward Selection (SFS) basado en la medida de FISHER [22]. Y el cuarto conjunto similar al tercero a diferencia que la SFS es basada en la medida KNN [22].

Se realizan pruebas de clasificación con técnicas de diferente naturaleza, a partir de estos conjuntos de datos, con una validación cruzada de 10-folds, obteniendo un desempeño superior al 90% para la mayoría de los conjuntos y clasificadores. Estos resultados iniciales demuestran la posibilidad de ampliar la investigación a la identificación del lenguaje de señas colombiano completo incluyendo los gestos dinámicos.

2. METODOLOGÍA

2.1 Base de datos

La base de datos para el reconocimiento de vocales en el lenguaje colombiano de señas fue obtenida con estudiantes del Instituto Tecnológico Metropolitano (ITM), donde a 151 personas adultas, entre los 18 y 50 años, se les indicó cómo realizar las cinco vocales en dicho lenguaje. Con un fondo uniforme se capturó el objeto de interés con una cámara Web, de resolución 1024x768 píxeles. Es importante resaltar que se tuvo en cuenta anexar a la base de datos una clase que corresponde a la clasificación de no vocal, estos ejemplos fueron obtenidos de la mano realizando otras posiciones, brazos, objetos, entre otros. Lo anterior lleva a obtener seis clases en total, es decir, las cinco vocales y una clase adicional de no vocal. En la Fig. 1. se muestran algunas imágenes ejemplo de la base de datos para las 6 clases consideradas.

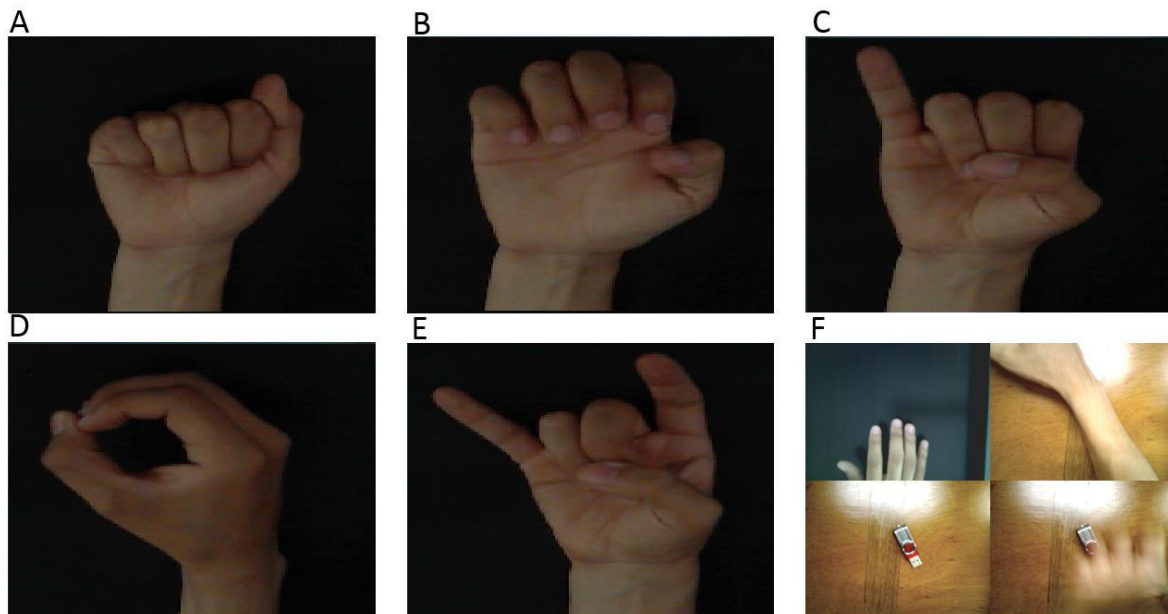


Fig. 1. Base de datos: (A) Clase 1 - vocal a, (B) Clase 2 - vocal e, (C) Clase 3 - vocal i, (D) clase 4 - vocal o, (E) clase 5 - vocal u, (F) clase 6 - no vocal. Fuente: Autores

Al conjunto de imágenes obtenidas en la base de datos se les realizó un preprocesamiento que constó de cuatro pasos. El primero de ellos fue aplicar un filtro gaussiano a la imagen original con el fin de suavizar y eliminar ruido en la imagen. El segundo fue una segmentación por color en el espacio de color Hue-Saturation-Value (HSV) que extrajo la totalidad de la mano al tener en cuenta los canales H y S, ya que son los que representan una variación importante para separar el color de la mano del fondo de la escena. El tercer paso realizado fue una operación de cierre para filtrar algunas regiones no pertenecientes al gesto, concluyendo el preprocesamiento con el último paso correspondiente a una umbralización por el método Otsu [23]. Como resultado a esta etapa se obtuvo un único objeto dentro de la escena, correspondiente a la mano.

Con la correcta segmentación de la mano, explicada anteriormente, se pudo realizar la extracción de características, para ello se redimensiona cada objeto co-

nexo a un tamaño de 60x90 píxeles (Fig. 2a.), del cual se adquirieron las características de: perímetro, centro en Y, amplitud, compacidad, diferencia en X, diferencia en Y, excentricidad, área, longitud eje menor, longitud, longitud eje mayor y momento.

Partiendo de la misma imagen segmentada y escalada, se divide en 54 zonas iguales de 10x10 píxeles. Cada zona representará una característica al recorrer las 19 líneas diagonales de la zona, promediando el número de píxeles de cada línea y luego promediando sobre las 19 líneas totales, lo cual está representado gráficamente en la Fig. 2b. Este es el método propuesto por [20], con el que se obtienen como resultado 54 características adicionales para un total de 66. Este primer grupo de características será llamado *Conjunto 1*. Cabe aclarar que todos los algoritmos de adquisición, preprocesamiento, segmentación y extracción de características se realizaron sobre librerías de código abierto OpenCV.

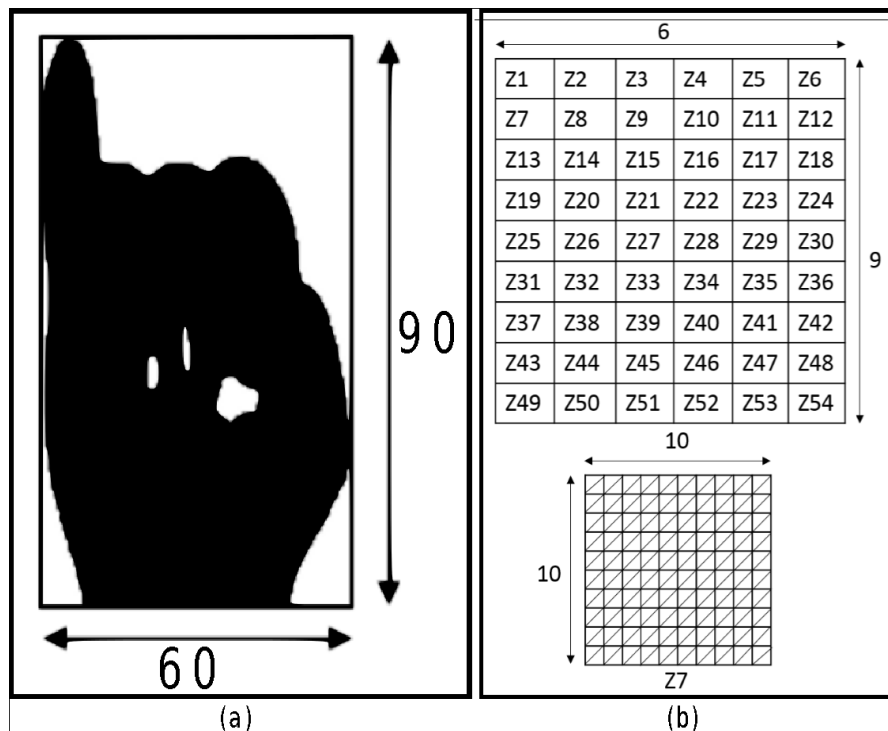


Fig. 2. (a) Redimensión de objeto conexo en 60X90 píxeles. (b) Extracción de las 54 características. Fuente: Pradeep et al. [20].

2.2 Marco experimental

Como se observa en la Fig. 3., la metodología abarca cuatro etapas principales. Se parte de la extracción de características (A), teniendo el *Conjunto 1* como resultado. Posteriormente se realiza una etapa de selección de características (B), de la cual se generan tres nuevos conjuntos de datos, ya sea mediante técnicas de selección o extracción de características. El *Conjunto 2* consta de d_{pca} nuevas características, obtenidas al aplicar el algoritmo de reducción de dimensión del análisis de componentes principales (PCA) sobre el *Conjunto 1*. Para generar el *Conjunto 3*, se selecciona un total de d_{fisher} características al tomar el *Conjunto 1* y aplicar Selección secuencial hacia adelante (SFS) basado en la medida de FISHER [24]. Y, por último, para generar el *Conjunto 4* se seleccionan un total de d_{knn} características aplicando SFS basado en el desempeño del clasificador KNN al *Conjunto 1*.

Teniendo estos conjuntos, cada uno es usado para entrenar un grupo de clasifica-

dores, los cuales son: SVM con los kernels de función de base radial (rbf), polinomial (poly), cuadrático (quad) y lineal (lin); KNN, Análisis Discriminante Lineal, Análisis Discriminante Cuadrático (QDA), Redes Neuronales Artificiales y Mahalanobis [25] [26]. Para la validación de los clasificadores considerados, se realizó una validación cruzada con 10 folds, etapa (C). Finalmente, en la etapa (D), se obtiene el desempeño de clasificación de cada uno de los modelos previamente entrenados con cada conjunto de características. Las etapas B, C y D de la Fig. 3. fueron desarrolladas con el Toolbox Balu de MATLAB [27].

3. RESULTADOS Y DISCUSIÓN

Según lo mencionado anteriormente, se obtienen cuatro conjuntos de características para la misma base de datos, con las que fueron entrenados clasificadores de diferente naturaleza, mostrando el desempeño de clasificación para cada uno.

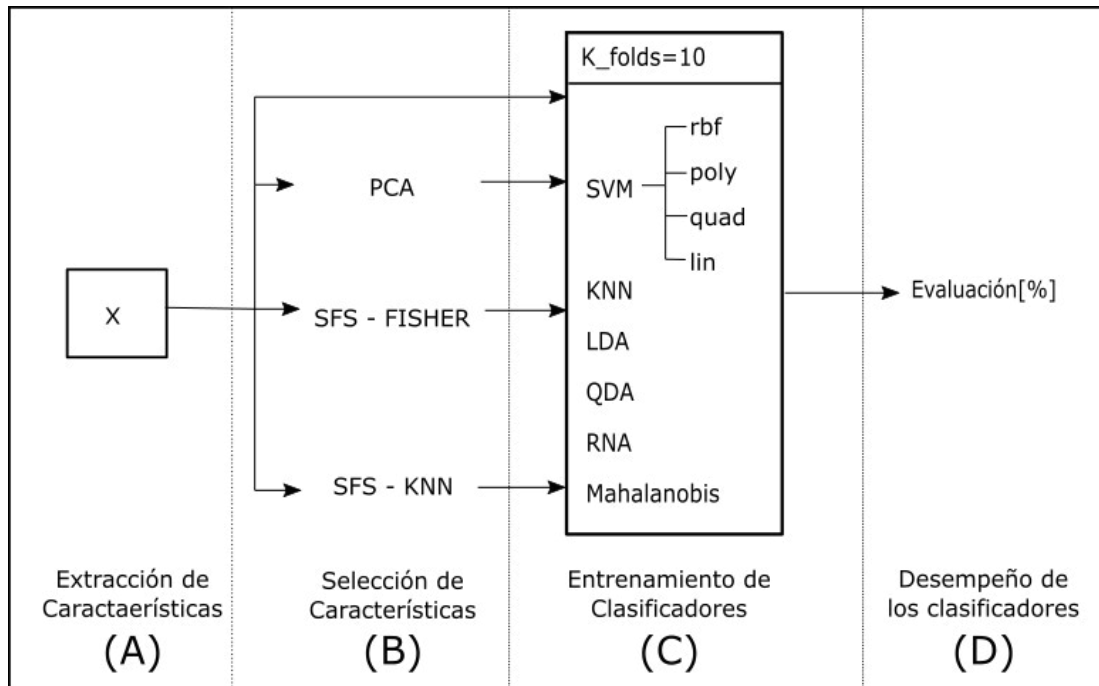


Fig. 3. Marco experimental: (A) Extracción de características, (B) Selección de características, (C) Entrenamiento de clasificadores, (D) Desempeño de los clasificadores. Fuente: autores.

Tabla 1. Desempeños de clasificación para el Conjunto 1 de datos. Fuente: autores.

Evaluación <i>Conjunto 1</i>	
Clasificador	Desempeño [%]
SVM-rbf	21,95 \pm 3,11
SVM-poly	95,69 \pm 1,52
SVM-quad	94,68 \pm 1,69
SVM-lin	94,7 \pm 1,68
KNN-k=7	36,33 \pm 3,61
LDA	22,4 \pm 3,13
QDA	3,22 \pm 1,32
RNA	35,19 \pm 3,58
Mahalanobis	15,78 \pm 2,74

3.1 *Conjunto 1*

La Tabla 1. muestra los resultados obtenidos al ingresar las 66 características del *Conjunto 1* por el grupo de clasificadores propuestos. El método QDA muestra el desempeño más bajo de todos. La mejor técnica para estos datos es el SVM-poly con un desempeño del 95,69% con desviación estándar de 1,52.

3.2 *Conjunto 2*

Para este conjunto de datos se usó PCA para retener el 99% de la varianza, lo cual da como resultado $d_{pca} = 28$ dimensiones o características.

La Fig. 4. representa el valor medio y la desviación estándar del desempeño de los clasificadores para el *Conjunto 2* de características planteado. KNN con $k=7$ muestra el desempeño promedio más bajo, mientras que el método QDA tiene el más alto con un 96,92% a 22 características.

3.3 *Conjunto 3*

d_{fisher} características al seleccionarlas mediante SFS-FISHER.

La Fig. 5. representa el valor medio y la desviación estándar del desempeño de los clasificadores para el *Conjunto 3* de características planteado. KNN con $k=7$ al igual que en la Fig. 3. mantiene el desempeño más bajo, mientras que el método que arrojó mejores resultados fue el LDA con 94.9% a 28 características.

3.4 *Conjunto 4*

d_{knn} características al seleccionarlas por SFS-KNN.

La Fig. 6. muestra el valor medio y la desviación estándar del desempeño obtenido al utilizar el *Conjunto 4* con los clasificadores para los 10 folds. Se puede observar cómo el KNN con $k=7$ sigue siendo el menos conveniente. Para esta evaluación en particular, el clasificador con mejor desempeño fue el SVM-rbf con 94,26% a 10 características.

KNN mostró ser uno de los peores clasificadores para este tipo de aplicación, ya que su desempeño es de los más bajos en la evaluación de la mayoría de conjuntos. Por otro lado, en la mayoría de evaluaciones, tres o cuatro clasificadores se encontraron en un desempeño mayor al 90%, indicando una buena predicción entre cada clase.

Finalmente, del análisis de los resultados con mejor desempeño usando selección de características, se escogió el método de selección que generó mejores resultados (SFS-Fisher) con su número de características (25), es decir, el *Conjunto 3*. A partir de estos datos se utilizó PCA desde 4 hasta 25 características, con pasos de tres, y se

evaluaron con todos los clasificadores. Los resultados fueron buenos, pero no mejores que usando QDA a 22 características para el *Conjunto 2* (que mostró ser el mejor clasificador para esta aplicación). Mejor resultado 94.49% con QDA para 16 características.

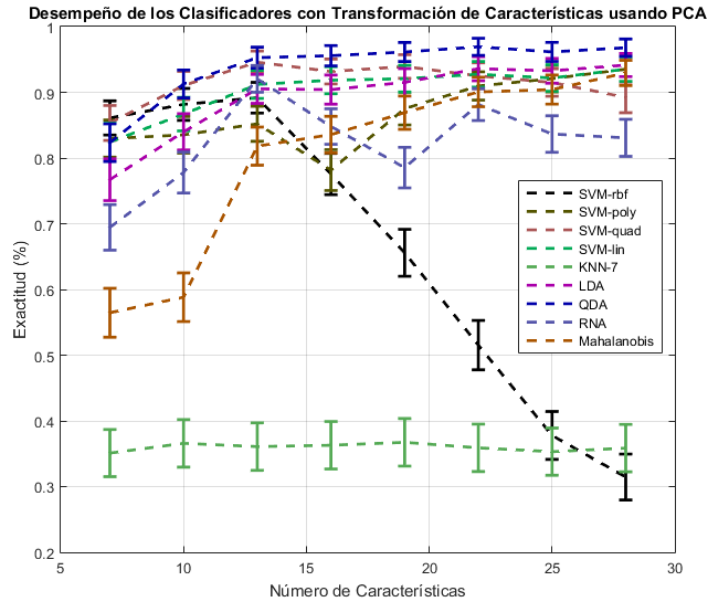


Fig. 4. Desempeño de los clasificadores para el *Conjunto 2* de características: 28 Características obtenidas por transformación de características usando PCA. Fuente: autores.

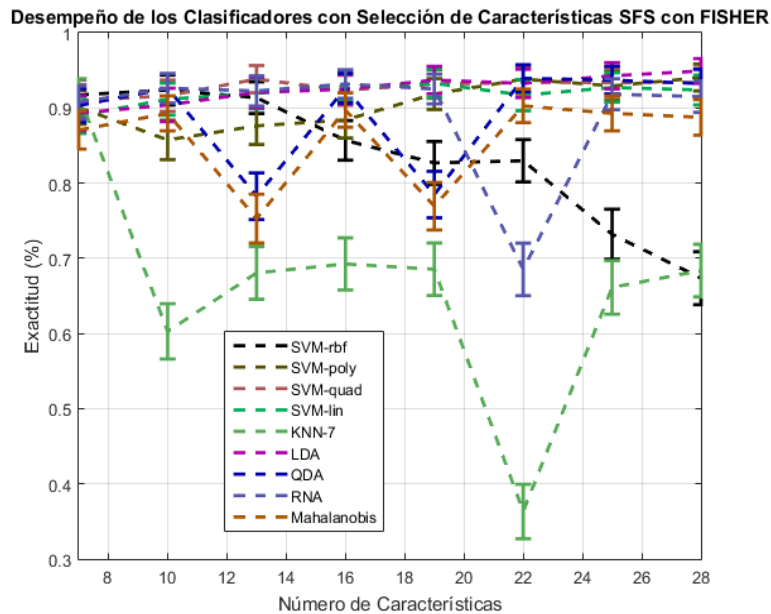


Fig. 5. Desempeño de los clasificadores para el *Conjunto 3* de características: Selección de características SFS con Fisher. Fuente: autores.

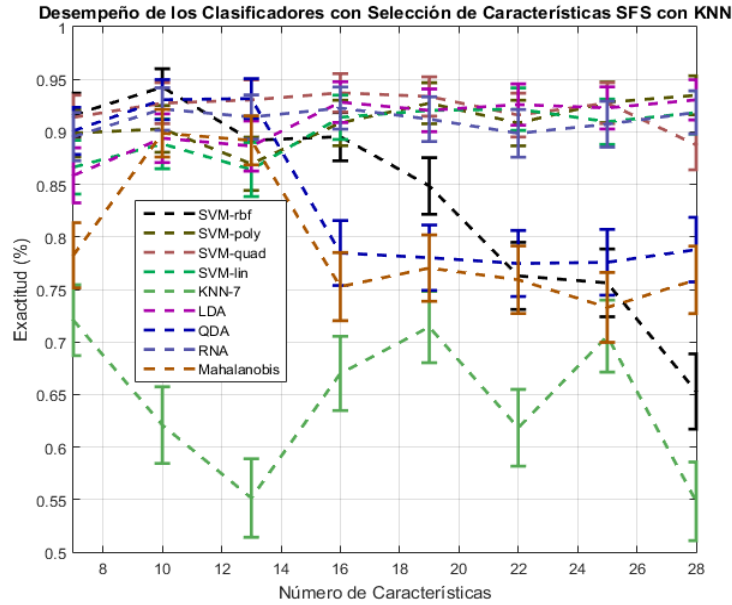


Fig. 6. Desempeño de los clasificadores para el *Conjunto 4* de características: Selección de características SFS con KNN. Fuente: Autores

4. CONCLUSIONES Y TRABAJOS FUTUROS

Se realizó una comparación de clasificadores, orientado a la identificación de las vocales del lenguaje colombiano de señas, junto a una clase extra correspondiente a ninguna vocal. Se creó una base de datos de 151 objetos de prueba para cada clase; se probaron cuatro conjuntos de características; el primero correspondiente a la totalidad de características propuestas en la metodología, el segundo por medio de la transformación de características usando PCA, llegando a un total de 28 características. El tercero y cuarto por SFS-FISHER y SFS-KNN, respectivamente; con un total de 28 características para ambos conjuntos.

Los resultados experimentales mostraron un mayor desempeño en la clasificación de las vocales del lenguaje colombiano de señas, con las características del *Conjunto 2: 28 Características por transformación de características usando PCA*. El mejor clasificador para este conjunto en particular fue el QDA a 22 características, obteniendo una tasa de desempeño del 96,92%, lo que

indica un excelente resultado para el problema planteado.

Como trabajos futuros se piensa generar una base de datos con un mayor número de clases, que abarque la totalidad del alfabeto en lenguaje colombiano de señas. También se puede investigar sobre el reconocimiento de señas dinámicas para poder identificar palabras y frases en el lenguaje colombiano de señas. La evaluación en otros clasificadores podría presentar buenos resultados; con un mayor número de clases y características, en un futuro se podría ampliar el área de investigación utilizando Deep Learning y Random forest como herramientas de clasificación. Con las mejoras obtenidas, se plantea generar una aplicación de software para reconocimiento en tiempo real de código abierto y que sea útil para la enseñanza y aprendizaje del lenguaje de señas colombiano.

5. REFERENCIAS

- [1] I. Instituto Nacional para Sordos, “Estadística Básica Población Sorda Colombiana,” 2015. [Online]. Available: <http://www.insor.gov.co/observatorio/estadisticas-basicas-poblacion-sorda-colombiana/>. [Accessed: 01-Sep-2017].
- [2] I. Instituto Nacional para Sordos, “Diccionario básico de la lengua de señas colombiana,” 2006. [Online]. Available: <http://www.ucn.edu.co/e-discapacidad/Documents/36317784-Diccionario-lengua-de-senas.pdf>. [Accessed: 27-Nov-2017].
- [3] A. V. W. Smith, A. I. Sutherland, A. Lemoine, and S. Mcgrath, “Hand gesture recognition system and method,” 6,128,003, 2000.
- [4] Y. Fang, K. Wang, J. Cheng, and H. Lu, “A Real-Time Hand Gesture Recognition Method,” in *Multimedia and Expo, 2007 IEEE International Conference on*, 2007, pp. 995–998.
- [5] P. Premaratne, S. Yang, Z. Zhou, and N. Bandara, “Dynamic Hand Gesture Recognition Framework,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8589 LNAI, 2014, pp. 834–845.
- [6] M. Hruz, J. Trojanová, and M. Železný, “Local Binary Pattern based features for sign language recognition,” *Pattern Recognit. Image Anal.*, vol. 22, no. 4, pp. 519–526, Dec. 2012.
- [7] Jie Huang, Wengang Zhou, Houqiang Li, and Weiping Li, “Sign Language Recognition using 3D convolutional neural networks,” in *2015 IEEE International Conference on Multimedia and Expo (ICME)*, 2015, pp. 1–6.
- [8] L. Pigou, S. Dieleman, P.-J. Kindermans, and B. Schrauwen, “Sign Language Recognition Using Convolutional Neural Networks,” in *Workshop at the European Conference on Computer Vision*, 2015, pp. 572–578.
- [9] G. Anantha Rao, P. V. V Kishore, A. S. C. S. Sastry, D. Anil Kumar, and E. Kiran Kumar, “Selfie Continuous Sign Language Recognition with Neural Network Classifier,” in *Lecture Notes in Electrical Engineering*, vol. 434, 2018, pp. 31–40.
- [10] X. Chai, G. Li, Y. Lin, Z. Xu, Y. Tang, and X. Chen, “Sign Language Recognition and Translation with Kinect,” *10th IEEE Int. Conf. Autom. Face Gesture Recognit.*, pp. 22–26, 2013.
- [11] K. M. Lim, A. W. C. Tan, and S. C. Tan, “A feature covariance matrix with serial particle filter for isolated sign language recognition,” *Expert Syst. Appl.*, vol. 54, pp. 208–218, Jul. 2016.
- [12] H. Cooper, B. Holt, and R. Bowden, “Sign Language Recognition,” in *Visual Analysis of Humans*, no. 231135, London: Springer London, 2011, pp. 539–562.
- [13] Eng-Jon Ong, H. Cooper, N. Pugeault, and R. Bowden, “Sign Language Recognition using Sequential Pattern Trees,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2200–2207.
- [14] A. Agarwal and M. K. Thakur, “Sign language recognition using Microsoft Kinect,” in *2013 Sixth International Conference on Contemporary Computing (IC3)*, 2013, pp. 181–185.
- [15] M. Mohandes, S. Aliyu, and M. Deriche, “Arabic sign language recognition using the leap motion controller,” in *2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE)*, 2014, pp. 960–965.
- [16] J. F. Lichtenauer, E. A. Hendriks, and M. J. T. Reinders, “Sign language recognition by combining statistical DTW and independent classification,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 11, pp. 2040–2046, 2008.
- [17] S. Dudoit, J. Fridlyand, and T. P. Speed, “Comparison of Discrimination Methods for the Classification of Tumors Using Gene Expression Data,” *J. Am. Stat. Assoc.*, vol. 97, no. 457, pp. 77–87, Mar. 2002.
- [18] M. Kumari and S. Godara, “Comparative Study of Data Mining Classification Methods in Cardiovascular Disease Prediction,” *Int. J. Comput. Sci. Trends Technol.*, vol. 2, no. 2, pp. 304–308, 2011.
- [19] M. Janidarmian, K. Radecka, and Z. Zilic, “Automated diagnosis of knee pathology using sensory data,” in *Proceedings of the 2014 4th International Conference on Wireless Mobile Communication and Healthcare - “Transforming Healthcare Through Innovations in Mobile and Wireless Technologies”*, MOBIHEALTH 2014, 2015, pp. 95–98.
- [20] J. Pradeep, E. Srinivasan, and S. Himavathi, “Diagonal Based Feature Extraction for Handwritten Alphabets Recognition System Using Neural Network,” *Int. J. Comput. Sci. Inf. Technol.*, vol. 3, no. 1, pp. 27–38, Feb. 2011.
- [21] C. Goodall and I. T. Jolliffe, “Principal Component Analysis,” *Technometrics*, vol. 30, no. 3, p. 351, Aug. 1988.
- [22] B. Al-Mistarehi, “An approach for automated detection and classification of pavement cracks,” 2017.

- [23] D. Liu and J. Yu, "Otsu Method and K-means," in *2009 Ninth International Conference on Hybrid Intelligent Systems*, 2009, vol. 1, pp. 344–349.
- [24] A. K. Jain, P. W. Duin, and Jianchang Mao, "Statistical pattern recognition: a review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 1, pp. 4–37, 2000.
- [25] S. Theodoridis, "Machine Learning: A Bayesian and Optimization Perspective," *Mach. Learn. A Bayesian Optim. Perspect.*, pp. 1–1050, 2015.
- [26] R. O. Duda, P. E. Hart, and D. G. Stork, "Pattern Classification," *New York John Wiley, Sect.*, p. 654, 2000.
- [27] D. Mery, "BALU: A Matlab toolbox for computer vision, pattern recognition and image processing", <http://dmery.ing.puc.cl/index.php/balu>, 2011.