

Histograma orientado a gradientes con máquina de soporte vectorial, en la clasificación del alfabeto dactilológico

Jhon Alejandro Intriago Chichanda^a, Fernando Rodrigo Moreira Moreira^a, Cinthia Mabel Sánchez Macías^a, Jéssica Johanna Morales Carrillo^a

^aCarrera de Computación, Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López, Campus Politécnico El Limón, Km 2.7, Calceta, Ecuador.
jessjohannamor@hotmail.com

Resumen. El objetivo del trabajo fue evaluar la eficiencia del clasificador Máquina de Soporte Vectorial (SVM) con el método de separación lineal, cuando se utiliza el algoritmo de extracción de características Histograma Orientado a Gradientes (HOG) y cuando no se utiliza ningún extractor de características en la clasificación del alfabeto dactilológico, variando en cada caso el parámetro de decisión del clasificador (C) que ajusta la frontera de decisión. Se utilizó el método bibliográfico para obtener los fundamentos teóricos y poder hacer un estudio de casos que fue experimentado mediante el método científico. Los resultados de esta experimentación demostraron que al utilizar HOG, se reduce los tiempos de entrenamiento y clasificación en un 75%, y a la vez se aumenta la precisión a un 6.97%, en comparación cuando se procede a clasificar sin un algoritmo de extracción de características. Además, en ambos casos, la precisión se mantiene estable a partir de la tercera iteración que corresponde a C igual a 100. Con esto se concluye que, con SVM es más conveniente utilizar el kernel lineal y HOG, por el tiempo y la precisión.

Palabras clave: Soporte vectorial, gradientes, kernel lineal.

1. Introducción

Machine learning o Máquina de aprendizaje, fue definida desde 1959 como un campo de estudio que da a las computadoras la capacidad de aprender sin ser programada de forma explícita (Phil, 2013). Además, Russell y Norving (2010) manifiestan que un computador puede mejorar su rendimiento en tareas futuras, después de hacer observaciones en el mundo real. Es usada para reproducir patrones conocidos y hacer predicciones basadas en los patrones. Se divide en aprendizaje supervisado y aprendizaje no supervisado.

El objetivo del aprendizaje supervisado es hacer predicciones basadas en comportamientos o características que se han visto en los datos almacenados, para poder buscar patrones relacionando todos los campos con el campo objetivo (González, 2014). Mientras que el aprendizaje no supervisado, según Russell y Norving (2010) es cuando el agente aprende los patrones de entrada sin necesidad de una retroalimentación explícita.

La extracción de características es una etapa importante en el procesamiento de imágenes (Nagarajan y Subashini, 2013). Los métodos basados en apariencia toman en cuenta la intensidad de los píxeles y características de la relación entre los valores de los píxeles, como el contraste entre regiones, bordes, gradientes, textura, entre otros (Cheung y Medina, 2013). Estas características se extraen para utilizarlas desde un modelo que permita clasificarlas automáticamente.

En este esquema se calculan Histogramas Orientados a Gradientes (HOG) en celdas de $N \times N$ píxeles. Estas celdas se agrupan en bloques y un descriptor es formado por el conjunto de histogramas concatenados (Merchán *et al.*, 2014). Para una mejor comprensión se ha construido un esquema con este procedimiento (Figura 1).

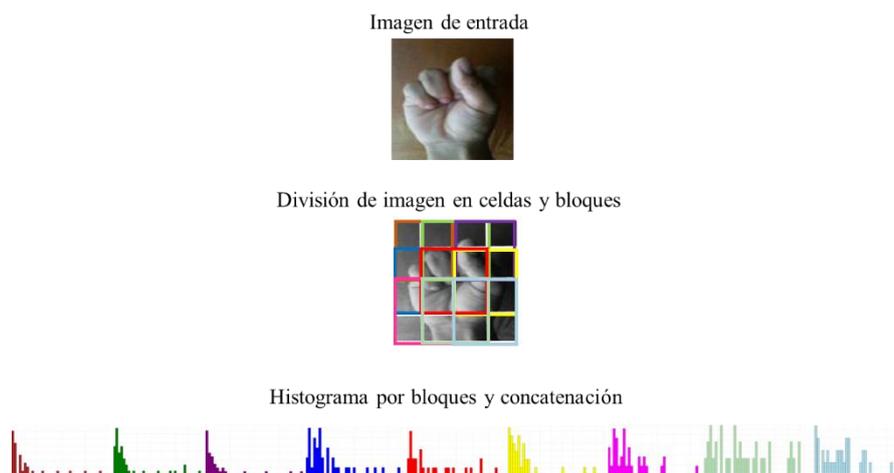


Figura 1. Histograma orientada a gradientes

HOG convierte la información local de los gradientes para cada pixel, en una representación de la imagen que captura la forma global del objeto en un vector de características. SVM (Support Vector Machine o Máquina de Soporte Vectorial) es uno de los algoritmos de clasificación más populares, basado en el aprendizaje estadístico que ha demostrado su eficiencia en muchas aplicaciones de reconocimiento de patrones (Han *et al.*, 2006). Este algoritmo tiene la capacidad de clasificar datos de alta dimensionalidad (Tarabalka *et al.*, 2010) y según Lee y Scott (2010) tiene métodos no paramétricos con gran alcance de estimación.

Se utiliza una función kernel para transformar los datos del espacio de entrada en un espacio de características de alta dimensional en la que se busca un hiperplano de separación, se tiene la función lineal, polinomial, RBF (Radial Basis Function), entre otros (Baitharu *et al.*, 2015).

Un dataset es un conjunto de datos conformado por las muestras con sus respectivas etiquetas o clases. Aguilar (2013) resalta que el dataset es importante para poder construir el modelo de clasificación, por medio del entrenamiento y validación del clasificador. La Figura 2 muestra los datos utilizados que corresponden al alfabeto dactilológico español, extraído del Glosario Básico de Lengua de Señas Ecuatoriana, por la FENASEC (2012).

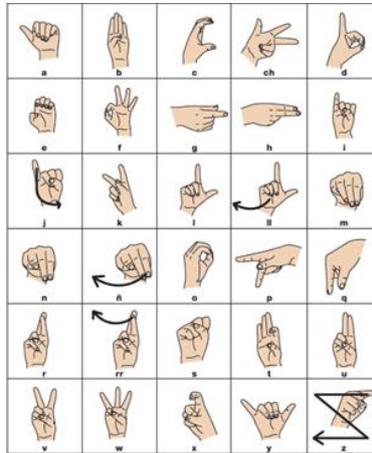


Figura 2. Alfabeto dactilológico

Con estas referencias se decidió experimentar con y sin HOG, para determinar los beneficios de la extracción de características, previa a la clasificación de imágenes estáticas del alfabeto dactilológico mediante SVM. Además, variando el valor del parámetro C, en cada caso.

2. Materiales y métodos

Método bibliográfico

El método bibliográfico fue utilizado para examinar trabajos relacionados, principalmente de artículos científicos, donde trataban temas acerca de visión por computador, extracción de características y algoritmos de entrenamiento supervisado.

Estudio de casos

Los casos estudiados fueron: SVM con el algoritmo de extracción de características HOG y SVM sin utilizar ningún algoritmo de extracción de características, con el fin de corroborar suposiciones y afirmaciones cualitativas plasmadas en varios documentos; y posteriormente poder experimentar con distintos datos.

Científico

Con los fundamentos obtenidos de la revisión bibliográfica, se procedió a analizar el algoritmo de extracción de características HOG (Histograma Orientado a Gradientes) utilizando el kernel lineal (método de separación de clases) y el algoritmo de clasificación SVM (Maquinas de Soporte Vectorial), para el reconocimiento de señas. Se realizó una comparativa de los resultados obtenidos con el algoritmo HOG y sin él, en el reconocimiento del alfabeto dactilológico ecuatoriano.

3. Resultados y discusión

Con base en la revisión bibliográfica se obtuvieron los fundamentos para escoger como algoritmo de extracción de características el Histograma Orientado a Gradientes (HOG), aunque también existen investigaciones donde no los utilizan. Además, se prefirió trabajar con una clasificación supervisada usando Soporte Vectorial de Máquina (SVM) con el método de separación de clases lineal.

En SVM existen parámetros de optimización, los principales son los ratios de aprendizaje (GAMMA) y los parámetros de penalización (C). Los valores por defecto de GAMMA se ajustan automáticamente y C es igual a uno; este último parámetro sirve para ajustar la frontera de decisión del clasificador, permitiendo personalizarla dependiendo del tipo de objeto a clasificar (alfabeto dactilológico).

Para seleccionar el parámetro de penalización y algoritmo de extracción de características más adecuado, se experimentó con 20 imágenes (100x120 pixels) por cada letra del abecedario dactilológico (26 clases), dando un total de 520 imágenes, más el resultado de dos filtros aplicados (variación contraste y brillo) a este total de imágenes, suman un total de 1560 imágenes. Dicho conjunto de datos se volvió a redimensionar a 32x32 pixeles mediante la librería scikit - image y se le aplicó un filtro Gaussiano con la librería scipy.

Los resultados ubicados en cada experimentación son el promedio de 7 ejecuciones del entrenamiento, para la cual se utilizó el 80% de las muestras y para evaluar el 20% restante del mismo conjunto de datos, que eran agrupados aleatoriamente en cada entrenamiento con la librería scikit - learn. Esta librería también fue utilizada para construir el modelo variando el parámetro C.

El equipo utilizado para esta evaluación fue una laptop I7, 8 GB de RAM, con SO Windows 8.1, con Python 2.7 y librerías externas. El Cuadro 1 contiene los resultados del tiempo, precisión y desviación estándar del modelo probando con siete valores de C, utilizando el método lineal sin una extracción previa de las características.

Se puede observar que a partir de la tercer iteración se mantiene la máxima precisión que es igual a 83.54% con una desviación estándar de 2.44%.

Cuadro 1. Reconocimiento de señas, lineal sin HOG

Iteración	C	Tiempo	Precisión	Desviación estándar
1	1×10^0	93.59 s	81.99%	2.97%
2	1×10^1	56.37 s	83.48%	2.62%
3	1×10^2	56.59 s	83.54%	2.44%
4	1×10^3	56.57 s	83.54%	2.44%
5	1×10^4	58.16 s	83.54%	2.44%
6	1×10^5	58.12 s	83.54%	2.44%
7	1×10^6	56.72 s	83.54%	2.44%

El gráfico 1 y 2 muestran el comportamiento del tiempo y la precisión frente a esta situación, donde se utilizó el método lineal sin un algoritmo de extracción de características. El tiempo se redujo notablemente en la segunda iteración, luego prácticamente se mantuvo constante, mientras que la precisión inició casi por el 82%.



Gráfico 1. Comportamiento de tiempo y precisión.

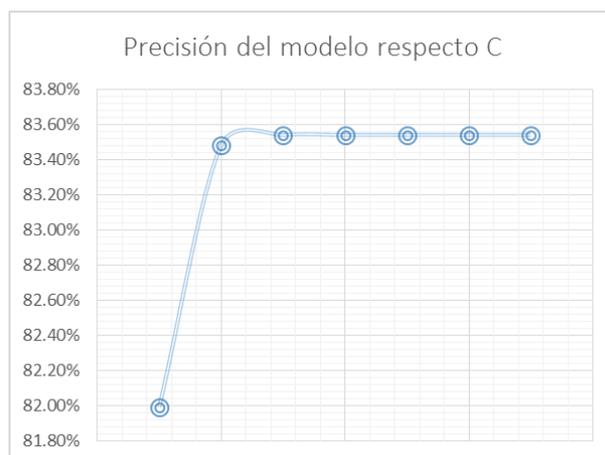


Gráfico 2. Reconocimiento de señas, linear sin HOG.

El Cuadro 2 contiene los tiempo, precisión y desviación estándar del modelo probando con siete valores de C, utilizando el algoritmo de extracción de características HOG y el método linear. Se puede observar que desde la tercer iteración se alcanza y mantiene la máxima precisión que es del 90.51% con una desviación estándar de 2.19%.

Cuadro 2. Reconocimiento de señas, linear con HOG

Iteración	C	Tiempo	Precisión	Desviación estándar
1	1×10^0	33.80 s	71.40%	2.03%
2	1×10^1	16.03 s	90.45%	1.77%
3	1×10^2	13.75 s	90.51%	2.19%
4	1×10^3	13.12 s	90.51%	2.19%

5	1×10^4	12.87 s	90.51%	2.19%
6	1×10^5	12.92 s	90.51%	2.19%
7	1×10^6	14.13 s	90.51%	2.19%

El Gráfico 3 y 4, muestra el comportamiento del tiempo y la precisión cuando se utilizó el algoritmo de extracción de características HOG y el método lineal. Este obtuvo los mejores resultados, menor tiempo en relación a la precisión obtenida.



Gráfico 3. Comportamiento de tiempo y precisión.

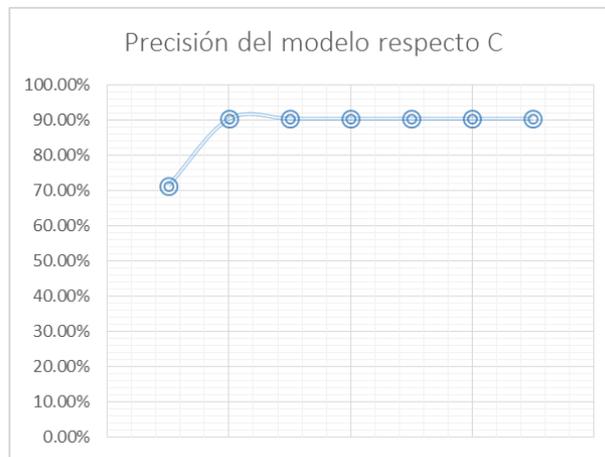


Gráfico 4. Reconocimiento de señas, linear con HOG

Si bien es cierto, en ambos casos el kernel lineal se estabiliza en la tercer iteración, pero con HOG mejora un 6.97% la precisión y disminuye el margen de error en un 0.25% con respecto a los resultados donde no se utiliza. Además de mejorar la precisión

se disminuyen notablemente los tiempos de entrenamiento en un 75% aproximadamente.

Soo y Bräunl (2015), a partir de los resultados de la evaluación realizada, encontraron que HOG superó a otra potente función llamada Gabor en los tres clasificadores evaluados. También indican que la combinación de la función de HOG y el clasificador SVM produce el mejor resultado.

4. Conclusiones

Los resultados de la experimentación controlando parámetros de SVM, permiten concluir que HOG mejora la precisión del algoritmo de clasificación y disminuye el margen de error, comparándolo con la clasificación sin utilizar previamente un algoritmo de extracción de características, demostrando que es más conveniente utilizar el kernel linear y HOG, por el tiempo y la precisión.

HOG obtiene un conjunto de características representativas de la imagen que facilita el entrenamiento del modelo, puesto que ya no requiere analizar todas las características de la imagen al momento de clasificar.

Referencias

1. Baitharu, T., Pani, S. y Dhal, S. 2015. Comparison of Kernel Selection for Support Vector Machines Using Diabetes Dataset. India. Journal of Computer Sciences and Applications. 3(6):181-184.
2. Cheung L. y Medina, C. 2013. Implementación y análisis de un detector de manos basado en visión artificial. Panamá. RIDTEC. 9(1):28-42.
3. FENASEC (Federación Nacional de Personas Sordas del Ecuador). 2012. Glosario Básico de Lengua de Señas Ecuatoriana. LSEC (Lengua de Señas Ecuatoriana).
4. González, A. 2014. Big data, data prediction, machine learning. Conceptos básicos de Machine Learning. Formato (HTML). Consultado el 10 de julio de 2016. Disponible en <http://cleverdata.io/conceptos-basicos-machine-learning/>.
5. Han, F., Shan, Y., Cekander, R., Sawhney, H. y Kumar, R. 2006. A two-stage approach to people and vehicle detection with hog-based SVM. In Performance Metrics for Intelligent Systems 2006 Workshop: 133-140.
6. Lee, G. y Scott, C. 2010. Nested Support Vector Machine. USA. IEEE. 58(3): 1648-1660.
7. Merchán, F., Galeano, S. y Poveda, H. 2014. Mejoras en el entrenamiento de esquemas de detección de sonrisas basados en AdaBoost. Panamá. RIDTEC. 10(2):17-30.
8. Nagarajan, S. y Subashini, T. 2013. Static Hand Gesture Recognition for Sign Language Alphabets using Edge Oriented Histogram and Multi Class SVM. India. IJCA (International Journal of Computer Applications). 82(4):28-35.
9. Phil, S. 2013. Too Big to Ignore: The Business Case for Big Data. Wiley. 72: 89.
10. Russell, S. y Norving, P. 2010. Artificial Intelligence. A modern approach. Pearson. 3 ed. p 695, 697.

11. Soo, T. y Bräunl, T. 2015. Performance Evaluation of HOG and Gabor Features for Vision-based Vehicle Detection. ICCSCE. IEEE. Penang, Malaysia. IEEE: 66-71.
12. Tarabalka, Y., Fauvel, M., Chanussot, J. y Benediktsson, J. 2010. SVM and MRF-Based Method for Accurate Classification of Hyperspectral Images. Francia. IEE. 7(4):736-740.