

ALEDO Algoritmo para la detección de objetos Algorithm ALEDO for the detection of objects

Leydi Mercedes Vargas Ordoñez¹, Luis Freddy Muñoz Sanabria², Francisco Javier Álvarez³

¹ Fundación Universitaria de Popayán, Sede Claustro San José Calle 5 No. 8-58, Cauca, Colombia, lady-117@hotmail.com

² Fundación Universitaria de Popayán, Sede Claustro San José Calle 5 No. 8-58, Cauca, Colombia, lfreddy@fup.edu.co

³ Universidad Autónoma de Aguascalientes, Av. Universidad # 940, Ciudad Universitaria, C. P. 20131, Aguascalientes, Ags, México, fjalvar@correo.uaa.mx

Fecha de recepción: 29 de junio de 2018

Fecha de aceptación: 8 de abril de 2019

Resumen. Esta investigación propone un algoritmo para la detección de objetos denominado ALEDO, para ello se hace una revisión en la literatura sobre los algoritmos más usados en estas actividades y se comparan contra ALEDO mediante modelos matemáticos. Se realiza una aplicación móvil usando ALEDO para el reconocimiento de señales de tránsito y basado en el concepto de aprendizaje supervisado. Al verificar la eficacia de ALEDO en la detección de imágenes frente a los algoritmos más usados para el mismo fin, dio como resultado que además de ser eficiente, cumplió con los propósitos para el que fue desarrollado. Es necesario continuar aplicando el algoritmo en otras aplicaciones móviles para la detección de imágenes que permitan solidificar los resultados.

Palabras clave: aprendizaje, supervisado, ALEDO, eficacia, detención de objetos, algoritmo.

Abstract. This research proposes an algorithm for the detection of objects called ALEDO, for which a review is made in the literature about the algorithms most used in these activities and they are compared against ALEDO through mathematical models. A mobile application is made using ALEDO for the recognition of traffic signals and based on the concept of supervised learning. When verifying the effectiveness of ALEDO in the detection of images in front of the most used algorithms for the same purpose, it resulted in that besides being efficient, it fulfilled the purposes for which it was developed. It is necessary to continue applying the algorithm in other mobile applications for the detection of images that allow to solidify the results.

Keywords: supervised learning, ALEDO, efficiency, stopping objects, algorithm.

1 Introducción

La humanidad al igual que muchas de las especies del mundo animal tienen la facultad de reconocer fácilmente diferentes objetos en imágenes con poco esfuerzo, sin importa las variantes en cuanto a escala, iluminación y ángulos de vista. No obstante, esta tarea es desafiante para los sistemas de visión computarizada. El mundo de la visión por computadora se deriva de la inteligencia artificial; cuyo propósito es entender una escena y las características de una imagen. La detección de objetos tiene un sinfín de aplicaciones en la actualidad en campos como la meteorología, la medicina, los sistemas de seguridad, la búsqueda y detección de patrones, entre otros [1].

El proceso de la detección de objetos se hace a través de un sistema inteligente capaz de extraer información de un entorno y luego implementar los algoritmos adecuados para su procesamiento y correcta interpretación, independientemente de factores como la iluminación, color, posición entre otros [2], y así reconocer todas las posibles variaciones que el objeto o imagen pueda representar.

Los algoritmos más usados para detección de objetos o imágenes son los descriptores, sin embargo, aún no existe un algoritmo cien por ciento capaz de responder de manera precisa a la gama de variaciones que puede afectar a una imagen [3], además consumen muchos recursos de CPU (unidad central de procesamiento) y memoria RAM (memoria de acceso aleatorio).

2 Estado del arte

Para el planteamiento de ALEDO, se hizo una revisión de la literatura científica sobre el reconocimiento de objetos y como llevar esta temática a cabo. Partiendo de lo investigado se determinó que el algoritmo de reconocimiento debe ser de aprendizaje automático [4]; el cual toma decisiones a partir de una cantidad de datos de entrenamiento disponible o pares de entradas y salidas deseadas. [5]

Cuando se aborda un problema de reconocimiento de objetos usualmente se sigue un proceso básico de cinco pasos que son: [6].

- Adquisición de la imagen: capturar la escena del mundo real a través de sensores y digitalizarla para su procesamiento, almacenamiento y transmisión,
- Pre procesamiento: aplicar técnicas de mejoramiento de contraste, reducción de ruido, realce de características, etc., de modo que la imagen se adecue para los siguientes pasos,
- Segmentación: aislar los objetos de interés de la imagen, extracción de rasgos: describir numéricamente la naturaleza de los objetos segmentados como su forma (color, textura, entre otros),
- Clasificación: asignar una clase o categoría a cada objeto de la imagen basado en sus rasgos. [7]

Para este proyecto se tuvo en cuenta algoritmos clasificadores ya que, estos permiten hacer representaciones de la información de una imagen de la siguiente manera: primero seleccionan puntos a describir, elaboran descripciones y luego comparan el modelo y la escena para obtener asociaciones, determinando si el objeto está presente y donde. [8] El clasificador Haar [9], determina a partir de una muestra si en la región es probable que este o no el objeto (arroja "1" si reconoce o "0" en caso contrario). Este clasificador está diseñado de tal modo que se puede cambiar el tamaño de detección con el fin de encontrar los objetos de interés en diferentes tamaños. Por lo tanto, para encontrar un objeto de un tamaño desconocido en la imagen el procedimiento de digitalización debe hacerse varias veces y a diferentes escalas.

El trabajo realizado por Alberto Pardo [10], quien para la automatización de robots que simulen el comportamiento humano e interactuar con el entorno, define un descriptor que reconocerá tanto el objeto como la escena donde está contenido y luego precisa un algoritmo clasificador que resuelve si el objeto hace parte de la colección de descripciones previas. En este proyecto se trató el problema de la detección de objetos en imágenes mediante técnicas de aprendizaje estadístico y visión por computador, se incluyó un filtro para detectar los posibles falsos positivos y eliminar detecciones erróneas similares al objeto a detectar, y finalmente se obtuvo una aplicación con un buen margen de detección.

Aleksandrov Ginchev [11] desarrolló una app llamada UDrive en la plataforma Android con la librería OpenCV para el tratamiento de imágenes específicamente para la detección de señales de tránsito, este proyecto logro con el tratamiento de imagen detectar características de una señal tales como el color, forma y caracteres numéricos, y el algoritmo resulto ser un buen punto de partida para detectar objetos, al reducir la resolución de la imagen de entrada, sin embargo simultáneamente reduce su eficacia para un posterior reconocimiento.

3 Metodología

Como caso de estudio se planteó desarrollar una aplicación móvil para el reconocimiento de señales de tránsito, ALEDO, utiliza como algoritmo clasificador a Haar, que toma muestras de ciertos puntos de un objeto en particular y los clasifica en muestras positivas (donde hay detección del objeto) y negativas (donde no hay detección del objeto), este clasificador además se caracteriza por hacer una detección en diferentes tamaños o escalas, por otro lado, ALEDO se apoyó en el algoritmo LBPH (Local Binary Patterns Histograms) que usa la idea básica de patrones binarios locales la cual resume la estructura local de una imagen mediante la comparación de cada pixel con su vecindario, principalmente este último se usa en el reconocimiento de rostros pero combinado con el clasificador Haar se vuelve más eficiente a la hora de detectar y reconocer objetos [13].

La aplicación móvil se desarrolló en la plataforma Android [3] con el apoyo de la librería OpenCV (librería de visión por computador de código abierto). Para el entrenamiento del clasificador se creó una muestra manual de 235 fotografías de señales de tránsito y para la muestra negativa se descargaron 200 imágenes de internet y con ayuda de la librería openCV [14] se hizo la conversión de estas muestras en un archivo de extensión txt.

A continuación, se muestran los fragmentos de código fuente donde se hace el entrenamiento del algoritmo descriptor y se programa ALEDO para la detección de las señales de tránsito [15].

Algoritmo 1. Entrenado al descriptor con las muestras positivas y negativas.

```
/Android_Proyecto/img/slObligatoria/img119.bmp (2) (442 120) (106 104) (587 158) (79 77)
opencv_createsamples -info slObligatoria.txt -num 235 -w 20 -h 20 -vec senal.vec
opencv_traincascade -data data -vec senal.vec -bg neg.txt -numPos 235 -numNeg 200 -numStages 15 -w 20 -h 20 -
featureType LBP
```

Algoritmo 2. Programación del algoritmo ALEDO para la detección de señal de tránsito en la situación de invariancia rotación.

```
public void entrenarSenal(){
    Mat m=new Mat();
    m=imgE.submat(r);
    Size s = m.size();
    Point centro = new Point(s.height/2,s.width/2);
```

```

if (countImages<MAXIMG)
{
    String aux= ".";
    int i1=-30;
    for (int i=0; i < 7; i++)
    {
        Mat rot_mat = Imgproc.getRotationMatrix2D(centro,i1, 1);
        Imgproc.warpAffine(m,m, rot_mat, m.size(), Imgproc.INTER_CUBIC);
        fr.add(m, text.getText().toString()+aux);
        aux=aux+".";
        countImages++;
        i1=10;
    }
}
}
}

```



Figura 1. Aplicación en ejecución, reconociendo una señal

4 Resultados preliminares

Se realizó un estudio comparativo entre ALEDO, y los descriptores ORB [16], SIFT [17], SURF[18] en tres situaciones reales para medir tres variables: escala, iluminación, y rotación de un objeto; que para este caso es una señal de tránsito que se detectó a 100 km/h y se hizo 10 pruebas por cada situación con el fin de tener pruebas más acertadas respecto a la eficiencia de ALEDO. A continuación, se muestran los resultados en milisegundos que cada algoritmo emplea para hacer la detección en diferentes escalas, situaciones de iluminación y ángulos de rotación de la señal de tránsito empleando Find_Object (una interfaz Qt simple para probar implementaciones OpenCV de SIFT, SURF, FAST, BRIEF y otros detectores y descriptores de funciones). [12]



Figura 2. Pantalla de Find Object con SURF

La interpretación de los datos arrojados por Find Object se tabulo en gráficos de barras para las tres situaciones (escala, iluminación, y rotación) a las que se sometieron los algoritmos descriptores.

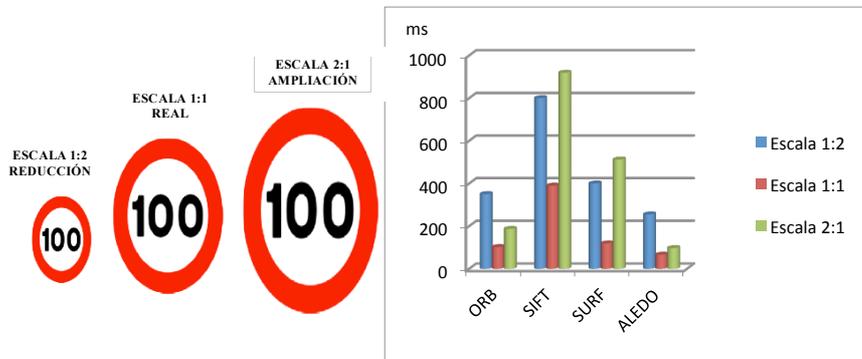


Figura 3. Situación 1: Grafica de comportamiento de detección en diferentes escalas.

Para la primera situación, la escala Find Object arrojó el tiempo en milisegundos que tardan los algoritmos en reconocer el objeto, para ello se hizo una variación de escalas en reducción, en escala real y en ampliación, en la gráfica se evidencia que para las diferentes escalas los algoritmos ORB, SIFT y SURFT reconocieron el objeto en rangos de tiempo comprendidos entre 300 y 800 milisegundos, pero ALEDO logró reconocer el objeto en un rango inferior a los 300 milisegundos, por lo tanto el algoritmo propuesto para esta primera prueba resultó ser más efectivo.



Figura 4. Situación 2: Grafica de comportamiento de detección en diferentes situaciones de iluminación.

Para la segunda situación, la iluminación Find Object arrojó el tiempo en milisegundos que tardan los algoritmos en reconocer el objeto, para ello se hizo una variación de iluminación en diferentes ángulos, en la gráfica se evidencia que para los diferentes ángulos de iluminación los algoritmos ORB, SIFT y SURFT reconocieron el objeto en rangos de tiempo comprendidos entre 200 y 500 milisegundos, pero ALEDO logró reconocer el objeto en un rango inferior a los 100 milisegundos, por lo tanto el algoritmo propuesto para esta segunda prueba resultó ser más efectivo.

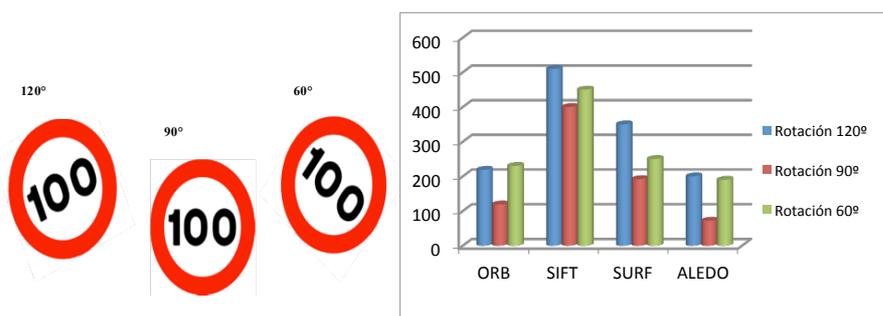


Figura 5. Situación 3: Gráfica de comportamiento de detección en diferentes ángulos de rotación.

Para la tercera situación, la rotación Find Objeto arrojó el tiempo en milisegundos que tardan los algoritmos en reconocer el objeto, para ello se hizo una variación de inclinación en diferentes ángulos, en la gráfica se evidencia que para los diferentes ángulos de inclinación los algoritmos OBR, SIFT y SURFT reconocieron el objeto en rangos de tiempo comprendidos entre 200 y 500 milisegundos, pero ALEDO logró reconocer el objeto en un rango inferior a los 200 milisegundos, por lo tanto el algoritmo propuesto para esta tercera prueba resultó ser más efectivo.

Finalmente este estudio comparativo se muestra claramente que ALEDO se desempeñó con mayor rapidez en las diferentes pruebas de detección. El desempeño del clasificador Haar [19] como detector y el algoritmo LBPH como descriptor para identificar las señales de tránsito fue exitoso, ALEDO obtuvo una gran capacidad de detección y reconocimiento de objetos en este caso una señal de tránsito distinguiendo la categoría a la que pertenece (preventiva u obligatorio), pero también obtuvo un notable porcentaje de reconocimiento en los ángulos entrenados. [20]

5 Conclusiones y trabajos futuros

El algoritmo propuesto en esta investigación, es altamente adaptable a diferentes aplicaciones. Al estimar la posición del objeto en la escena, permitió sobreponer información virtual específica [21].

La revisión de la literatura permitió elaborar un conjunto de métodos y una correcta modificación de los algoritmos para llevar a cabo el reconocimiento de señales de tránsito en escenas controladas, estableciendo con ello una metodología para llevar a cabo la detección del objeto a través de un aprendizaje supervisado [22].

Como trabajo futuro se propone hacer un estudio sobre el clasificador Haar, ya que los datos de entrenamiento deben ser aproximados a 5000 imágenes positivas y negativas para reducirlo y mejorar su alcance o distancia de detección, aspecto que en Haar debe mejorarse.

Por otro lado también deben realizarse otros estudios donde se implemente el algoritmo Haar con otros descriptores y así ALEDO se potencializará para ser usado en diferentes áreas del conocimiento [20].

Referencias

- [1]: Ku, M. C. B., Jiménez, D., SLP, D., & Hayet, J. B. (2014). Reconocimiento de objetos en escenas complejas para aplicaciones de realidad aumentada. Museo.
- [2]. Shih, F. Y. (2010). Image processing and pattern recognition: fundamentals and techniques. John Wiley & Sons.
- [3]. Ibay, Ñ., Fabricio, M., Córdova, T., & Gonzalo, L. (2013). Análisis de eficiencia en algoritmos de reconocimiento de imágenes digitales aplicables a dispositivos móviles bajo la plataforma Android (Bachelor's thesis, Universidad de las Fuerzas Armadas ESPE. Carrera de Ingeniería en Sistemas e Informática.).
- [4]. Sierra Araujo, B. (2006). Aprendizaje automático: conceptos básicos y avanzados: aspectos prácticos utilizando el software Weka. Madrid: Pearson Prentice Hall.
- [5]. Molina, A. I., Redondo, M. A., & Ortega, M. (2003). Un método semiautomático basado en algoritmos genéticos para el análisis de experiencias de aprendizaje colaborativo. Taller en Sistemas Hipermedia Colaborativos y Adaptativos (JISBD'03).
- [6].González, A., Martínez, F., Pernía, A., Alba, F., Castejón, M., Ordieres, J., & Vergara, E. (2006). Técnicas y algoritmos básicos de visión artificial. UNIVERSIDAD DE LA RIOJA, SERVICIO DE PUBLICACIONES.
- [7]. Domínguez Torres, A. (1996). Procesamiento digital de imágenes. Perfiles Educativos, (72).
- [8]:Ahmed, M. T., & Amin, S. H. (2015). Comparison of face recognition algorithms for human-robot interactions. Jurnal Teknologi, 72(2), 1-6.
- [9].Wilson, P. I., & Fernandez, J. (2006). Facial feature detection using Haar classifiers. Journal of Computing Sciences in Colleges, 21(4), 127-133
- [10]. Pardo, A. E. (2009). Reconocimiento de objetos. Universidad de Barcelona, Barcelona.
- [11]. Ginchev, A. (4ª edición, julio - noviembre 2013). Proyecto final de curso Android: Programación de aplicaciones UDrive. <http://www.dcomg.upv.es/~jtomas/android/proyectos4ed/UDrive.pdf>

- [12]. Roth, P. M., & Winter, M. (2008). Survey of appearance-based methods for object recognition. Graz University of Technology, (Inf. Téc) Austria: [s.n.].
- [13]. Watts, A. C., Perry, J. H., Smith, S. E., Burgess, M. A., Wilkinson, B. E., Szantoi, Z. (2010). Small unmanned aircraft systems for low-altitude aerial surveys. *The Journal of Wildlife Management*, 74(7), 1614-1619.
- [14]. Mateu García, Ó. (2009). Análisis y detección de objetos de primer plano en secuencias de video.
- [15]. Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. Ponencia presentada en *Computer Vision and Pattern Recognition*, USA.
- [16]. Yu, L., Yu, Z., & Gong, Y. (2015). An improved ORB algorithm of extracting and matching features. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 8(5), 117-126.
- [17]. Flores, P., & Braun, J. (2011). Algoritmo SIFT: fundamento teórico.
- [18]. Huijuan, Z., & Qiong, H. (2011, September). Fast image matching based-on improved SURF algorithm. In *Electronics, Communications and Control (ICECC), 2011 International Conference on* (pp. 1460-1463). IEEE.
- [19]. Ball, T. (2013). Train Your Own OpenCV Haar Classifier. Recuperado 23 de Junio 2016, de <http://coding-robin.de/2013/07/22/train-your-own-opencv-Haar-classifier.html>
- [20]. Ross, J. A. (2008). Computer vision and target localization algorithms for autonomous unmanned aerial vehicles. *Dicertación Doctoral*, The Pennsylvania State University.
- [21]. Reinius, S. (2013). Object recognition using the OpenCV Haar cascade-classifier on the iOS platform [formato electrónico] [s.l.]: [s.n.].
- [22]. Branch, J., & Olague, G. (2001). La visión por computador: Una aproximación al estado del arte. *Revista Dyna*, 133.