

Sistema de visión artificial para el reconocimiento de productos en banda transportadora

Juan David Ospina¹ y Manuela Quiroz²

ABSTRACT *Abstract*—La detección y el reconocimiento de objetos, son dos de las tareas más recurrentes a las cuales los sistemas de visión artificial acceden para el desarrollo de múltiples aplicaciones tecnológicas, por ello y gracias al potencial de estos sistemas consecuencia de la innovación científica que se está viviendo en los últimos años, estas áreas de investigación se constituyen actualmente como asignaturas fundamentales para la competencia en cualquier ámbito ingenieril. Por ello y con el ánimo de continuar aportando al crecimiento de estas disciplinas, en el presente artículo se expone un sistema de visión artificial para la clasificación de productos en una banda transportadora a través de un algoritmo de OpenCV desarrollado en Python. Dicho lenguaje de programación debido a su simplicidad en la sintaxis y a la alta compatibilidad con diversas librerías de licencia libre, permite desarrollar grandes aplicaciones de visión y procesamiento de imágenes por computadora de manera relativamente simple. Para el alcance de dicho objetivo, se diseña un SVM (Support Vector Machine) como herramienta para la clasificación de productos. Su funcionamiento, permite utilizar un conjunto de muestras como entrenamiento para la svm que realizará la predicción de las etiquetas de clasificación de cada uno de los productos que se desplazan sobre la banda transportadora. Para facilitar el conteo y el rendimiento del método, se integra una GUI que permita monitorear de forma ágil los resultados obtenidos con el algoritmo de trabajo. Como es de esperarse, los resultados de este sistema no poseen una exactitud completa producto a la cantidad de interferencias del contexto (puntos de vista, oclusión, escala) y ruidos técnicos que no hemos tenido en cuenta, sin embargo, se ha alcanzado valores de clasificación bastante importantes, alcanzado en algunas ocasiones valores de 100 de exactitud en la matriz de confusión del svm (resultado obtenido al utilizar histogramas de 3 bins para RGB y HSV).

I. INTRODUCCIÓN

La visión artificial es una disciplina científica que tiene por objetivo la adquisición y procesamiento de imágenes del entorno para su posterior análisis y estudio. En base al desarrollo tecnológico de las últimas décadas, esta disciplina ha crecido sustancialmente alcanzando importantes logros en diversas áreas de investigación, como el reconocimiento de objetos, la navegación autónoma y la reconstrucción de modelos 3D, entre otros, creando nuevos entornos de desarrollo para todo tipo de aplicaciones tecnológicas. Desde el surgimiento del fenómeno de procesamiento de imágenes por computadora hasta el día de hoy, un número importante de librerías de visión artificial y entornos de desarrollo para

el procesamiento y análisis de estos datos han aparecido para su uso y comercialización, sin embargo, OpenCV (librería desarrollada por Intel) se ha posicionado como una de las bibliotecas de visión artificial de mayor uso en la actualidad, debido a su licencia de uso libre y a la alta compatibilidad que posee con los principales lenguajes de programación para el desarrollo, tales como C y C++, además de poseer extensiones de estos para lenguajes de más fácil indexación como Python, facilitando la accesibilidad y el rango de exploración para muchos investigadores atraídos por el diseño de aplicaciones con sistemas avanzados de adquisición y procesamiento de imágenes.

II. DEFINICIÓN DEL PROBLEMA

En el presente artículo se expone el diseño de un sistema de visión por computadora cuyo objetivo es la detección, identificación y conteo de 5 tipos de empaque de chocolatina que viajan a través de una banda transportadora, utilizando para ello un algoritmo con interfaz gráfica desarrollado en Python, compatible con OpenCV y articulado a una webcam Logitech C920 que está situada en la parte superior de una banda transportadora.

III. METODOLOGÍA

Para el diseño del sistema de visión artificial, se utilizará la librería OpenCV como base de trabajo en complemento con un algoritmo de aprendizaje supervisado tipo SVM (Support vector machine)

A. OpenCV

OpenCV es una biblioteca multiplataforma de visión artificial de licencia libre desarrollada por Intel. Contiene más de 500 funciones con aplicaciones de visión, reconocimiento, calibración de cámaras y robótica.

B. SVM

Las máquinas de soporte vectorial son un conjunto de algoritmos de aprendizaje supervisado desarrolladas por Vladimir Vapnik, las cuales se utilizan para la predicción de clases en un conjunto determinado de datos. Su funcionamiento básico consiste en tomar un conjunto de ejemplos de entrenamiento o muestras que podemos etiquetar manualmente para entrenar una SVM y construir un modelo basado en hiperplanos de espacio que permitan predecir la etiqueta de muestras nuevas de la forma más óptima posible.

*This work was not supported by any organization

¹Juan David Ospina Alcaraz es ingeniero mecánico de la Universidad Pontificia Bolivariana y actualmente es estudiante de la especialización en diseño integrado de sistemas técnicos de la universidad EAFIT, Colombia, Medellín jdospinaa@eafit.edu.co

²Manuela Quiroz es ingeniera física de la universidad EAFIT, Colombia, Medellín b.d.researcher@ieee.org

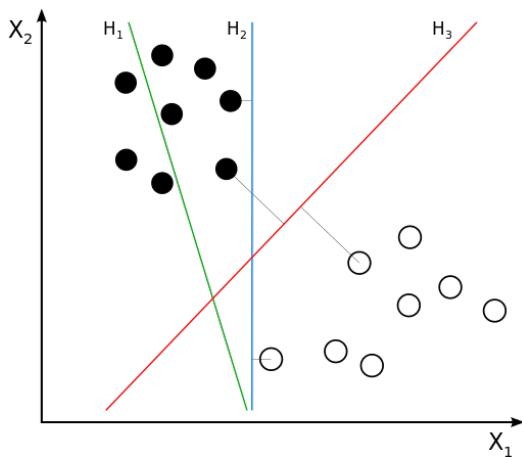


Fig. 1: La SVM genera y busca posibles hiperplanos que separen de mejor forma los puntos de una clase de las otras.

C. Entrenamiento SVM

El diseño del sistema comienza con el entrenamiento del svm:

- Primero capturamos las fotografías que utilizaremos para el entrenamiento del svm, en este caso tomaremos 50 imágenes de cada tipo de empaque de chocolatina y otras 50 de la banda sobre la cual se moverán.
- Luego aplicaremos una serie de filtros para facilitar el análisis de la información relevante en cada fotografía, comenzando por un recorte de píxeles de áreas de poco interés, como lo son los soportes de la banda transportadora, un filtro de suavizado (cv2.medianblur) y filtros para el acondicionamiento de los bordes (cv2.Canny, cv2.morphologyEx) los cuales conformarán la estructura básica de la máscara aplicada a las imágenes.



Fig. 2: Fotografía de chocolatina jumbo flow blanca luego de ser procesada

- Posteriormente, se crean los arreglos de las etiquetas y los atributos que serán reordenados aleatoriamente para prepararlos para el test del svm. Los atributos escogi-

dos en este caso son los valores del histograma para cada canal, tanto en RGB como en HSV, calculándolo únicamente para 3 bins.

- Utilizaremos el 90% para el entrenamiento de la svm y el 10% restante serán utilizados como test.
- En este momento, creamos nuestro vector de trabajo (cv2.ml.SVMcreate) que realizará pruebas aleatorias de etiquetado y evaluación, retroalimentándose por el arreglo que ha sido previamente etiquetado por el usuario. De este procedimiento obtenemos el dato de exactitud (accuracy) y el resultado del test en una matriz de confusión. Es posible realizar la cantidad de pruebas que se desee hasta obtener un svmdata satisfactorio para el trabajo que se desea realizar.

D. Desarrollo de la interfaz

- Utilizando la interfaz gráfica por defecto que posee el lenguaje Python (Tkinter) se diseñó una GUI que permita observar la detección de los diversos empaques de chocolatinas además de un contador de tiempo total de detección. La GUI además, condiciona por medio de un botón de inicio (start) el comienzo de trabajo de la webcam y la activación de la banda transportadora para empezar con el proceso de detección y etiquetado.
- con la instrucción v2.VideoCapture obtenemos las imágenes procedentes de la banda transportadora en tiempo real, aplicamos un seguimiento o tracking para detectar cuando un nuevo objeto entra en el frame de trabajo definido y aplicamos los mismos filtros que aplicamos a las muestras en el entrenamiento de la svm para las imágenes obtenidas de la webcam, esto con la intención de realizar una nueva predicción de etiquetado con la ayuda del svmdata obtenido anteriormente. En base a la exactitud, a los rangos de detección establecidos, a las condiciones de operación del sistema de la banda transportadora y a las condiciones de luz en la adquisición de las imágenes el nivel de precisión de clasificado de productos se verá afectado. La robustez de los parámetros y el sistema svm podría minimizar muchas de estas características del entorno y el sistema.

IV. RESULTADOS

A. Dataset

Teniendo el archivo .dat que contiene la SVM es posible emplearlo para predecir la clase de un objeto en un video en vivo, sin embargo, es de gran importancia evaluar su desempeño para imágenes estáticas. A continuación se describe el conjunto de datos usado para el entrenamiento y la evaluación del algoritmo de machine learning para clasificación.

Se tomaron 50 fotos para cada una de las chocolatinas y 50 fotos para la banda vacía, teniendo un total de 300 imágenes. Como se mencionó anteriormente, únicamente el 90% fue usado para entrenar el algoritmo, el 10% restante se usó para evaluar su desempeño. Entre las fotos se procuró incluir diferentes ángulos de posicionamiento, condiciones de iluminación e introducir deformaciones distintas al empaque.



Fig. 3: Algunas imágenes empleadas para el entrenamiento de la SVM

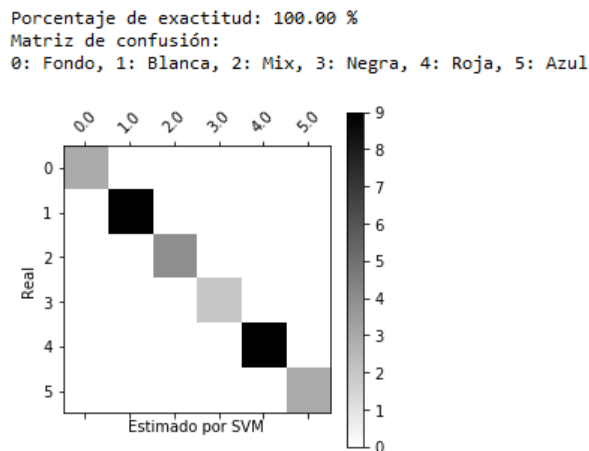


Fig. 4: Matriz de confusión para SVM entrenada con histogramas de 3 bins para RGB y HSV.

Se evaluó la posibilidad de emplear una gran variedad de atributos diferentes, por ejemplo, el área del objeto, la excentricidad o el histograma de gradientes orientados (HOG). Sin embargo, los resultados no superaron nunca el 30% de exactitud, principalmente porque las características geométricas de los objetos que se quieren clasificar no entregan tanta información como su composición de colores.

B. Matriz de confusión

Tras evaluar el desempeño de la svm entrenada, se obtuvieron los resultados que se observan en la figura 4 donde se observa que la exactitud es del 100% lo cual concuerda con el hecho de que todos los valores de la matriz de confusión se encuentren en la diagonal principal.

C. Tasa de clasificación

Durante las pruebas realizadas en vivo se emplearon diferentes archivos .dat obteniendo diferentes resultados. Era bastante común que la máquina confundiera la roja con la mix y la azul con la blanca. Sin embargo, se logró satisfactoriamente contar cada una únicamente una vez y hacerle tracking a cada una como se observa en la figura

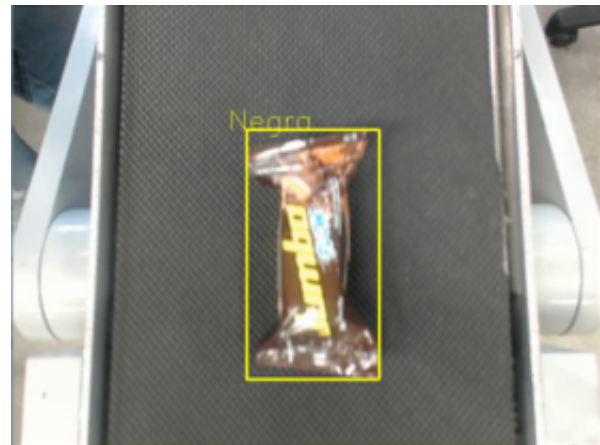


Fig. 5: Identificación de chocolatinas en tiempo real.

5. De igual forma, la clasificación predicha por la svm se mostraba en tiempo real sobre la chocolatina.

Esta discrepancia entre las predicciones con imágenes y con video en vivo se debe principalmente a cambios en la iluminación y los paros inesperados que tenía la banda transportadora por fallas mecánicas. Es posible mejorar la exactitud del algoritmo de predicción al usar una mayor cantidad de imágenes de entrenamiento junto con un arreglo de atributos más extenso. Sin embargo, para la clasificación de este tipo de elementos que difieren tanto en sus características de color, es innecesario la implementación de algoritmos más robustos y entrenados con datasets más grandes para obtener mejores resultados.

BIBLIOGRAFÍA

- Mallick, S. (2017). Handwritten Digits Classification : An OpenCV (C++ / Python) Tutorial — Learn OpenCV. Learnopencv.com. Retrieved 3 March 2018, from <https://www.learnopencv.com/handwritten-digits-classification-an-opencv-c-python-tutorial/>
- OpenCV: OpenCV modules. (2018). Docs.opencv.org. Retrieved 3 March 2018, from <https://docs.opencv.org/3.3.1/index.html>
- Policy, P., Vision, C., Network, N., and Policy, P. (2017). OpenCV Object Tracking by Colour Detection in Python - The Codacus. The Codacus. Retrieved 3 March 2018, from <https://thecodacus.com/opencv-object-tracking-colour-detection-python/#.WppSCehubIU>