Implementación de un sistema de detección de troncos de árboles utilizando histogramas de gradientes orientados (HOG)

Guillermo Steiner[a], Alvaro Shapiro, Eduardo Destefanis
[a]gsteiner@scdt.frc.utn.edu.ar

Centro de Investigación en Informática para la Ingeniería (CIII)
Universidad Tecnológica Nacional - Facultad Regional Córdoba

Resumen Se realiza un implementación de detección de troncos de arboles para su aplicación en una plataforma móvil empleando el algoritmo de Navneet DALAL en la extracción de características comúnmente usado en la detección de personas. Se plantea el modelo de entrenamiento utilizado para obtener los clasificadores mediante maquinas de vector de soporte (SVM) y se realiza una evaluación de los resultados obtenidos para diferentes tamaños de ventana.

SVM HOG

1. Introducción

La utilización de visión artificial en la navegación robótica, permite extraer información relevante del entorno, procesarla en tiempo real y a partir de esto tomar decisiones. Aplicando técnicas de aprendizaje se puede obtener un modelo robusto para su implementación en entornos reales. Una Máquina de Soporte Vectorial (SVM, por su siglas en ingles de Support Vector Machine) permite clasificar los objetos de la escena percibida pertenecientes a cierta clase de interés, por ejemplo troncos de árboles. La SVM es un método de aprendizaje supervisado que utiliza un conjunto de datos de entrenamiento de dos clases diferentes. Su principal objetivo es encontrar una adecuada hipótesis que permita separar con un margen máximo las dos clases de objetos. Para ello, se realiza un entrenamiento en base a muestras positivas y negativas de los objetos de interés, en el caso de este trabajo troncos de árboles, para luego aprender un modelo general de dicho objeto.

Esto se hace mediante la transformación de la imagen en un espacio de características mas adecuado, donde la información se representa en una forma más compacta y resulta mas sencillo definir el descriptor. Esta representación de los objetos en el espacio de las características se conoce como descriptor; siendo el Histograma de Gradientes Orientados (HOG, por sus siglas en inglés de Histogram of Oriented Gradients) [2] uno de los más utilizados actualmente.

La aplicación de este método, es para la navegación de un robot móvil en un ambiente agrícola. La misma será implementada en una plantación de olivos

perteneciente al Instituto Nacional de Tecnología Agropecuaria en San Juan. Se detectaron los troncos de los árboles para obtener la posición de los mismos en el ambiente y realizar de esa forma un mapa del mismo.

El trabajo se organiza de la siguiente manera. La sección 2 muestra la arquitectura general que fue implementada para obtener un modelo de detección. Esta etapa abarca la selección de imágenes, entrenamiento y aprendizaje del clasificador y el proceso de detección de árboles en la implementación. Realizando en cada paso una introducción teórica de los métodos utilizados. La sección 3 explica la metodología usada en la selección del tamaño de la ventana y La sección 4 muestra los resultados obtenidos para diferentes tamaños de ventanas. Por último se presentan las conclusiones del trabajo.

2. Arquitectura general

La arquitectura para la detección de troncos de árboles en regiones dentro de una imagen se divide en tres etapas

- *Creación de un conjunto de imágenes:* Se dividen en dos tipos:
  - Imágenes para entrenamiento y verificación, las cuales tienen que ser positivas (Árbol) y negativas (No Árbol).
  - Imágenes para prueba, de las cuales obtendremos los resultados del entrenamiento.
- *Fase de entrenamiento:* crea un clasificador que provee decisiones árbol/no-árbol para una región de la imagen de tamaño fijo (Figura 1a).
- *Fase de detección:* Utiliza el clasificador obtenido para buscar en imágenes árboles en todas las posiciones y escalas (Figura 1b) .

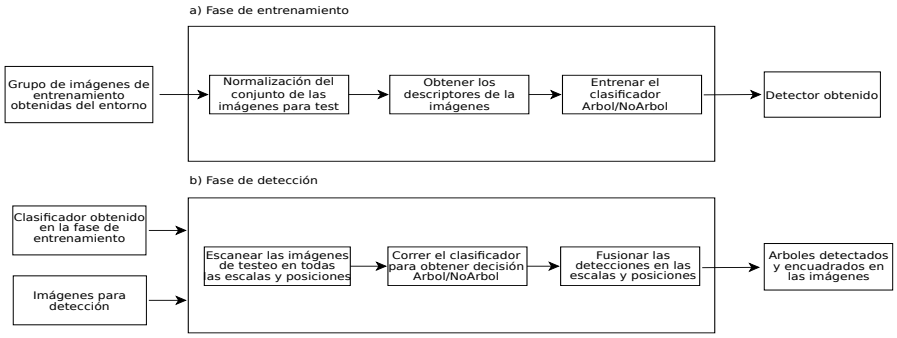

Figura 1. Etapas de entrenamiento y detección implementadas para la detección de árboles

2.1. Creación del conjunto de imágenes.

Obtención del conjunto de imágenes Debido a la necesidad que la detección de troncos de árboles sea aplicada al mundo real, el conjunto de imágenes seleccionadas tienen que ser representativas al entorno donde se desarrollará la aplicación. Debido a esto se descartan las imágenes cinéticas.

Para obtener el conjunto de imágenes, se filmaron varios vídeos con una cámara personal en el lugar donde se realizaría la implementación. De estos vídeo fueron extraídos los cuadros para realizar una selección del conjunto de imágenes positivas y negativas.

Figura 2. Imágenes de dimensión 640x480 píxeles del entorno donde se implementa el algoritmo de detección

De los cuadros obtenidos, se etiquetaron manualmente 508 imágenes de troncos de árboles con un poco de follaje. Debido a que el tamaño de los troncos varían, los mismos fueron luego escalados a un tamaño de 64x128 [1] píxeles y centrados. Estas imágenes formaron el grupo de *imágenes positivas*. Para aumentar este conjunto las mismas se reflejaron horizontalmente, por lo que se obtuvo una cantidad total de 1016 imágenes positivas (Figura 3).

Las imágenes de entrenamiento etiquetadas como *imágenes negativas,* fueron tomadas desde dos conjuntos de imágenes:

- 325 imágenes negativas formadas por las porciones etiquetadas manualmente de imágenes del entorno de la plantación de olivos que no poseían troncos de arboles.
- 590 imágenes del conjunto de entrenamiento para búsqueda de personas INRIA [5][3]. (De este conjunto de entrenamiento fueron eliminadas manualmente las imágenes que poseían troncos de árboles.)

De estos dos conjuntos de imágenes fueron tomadas porciones seleccionadas aleatoriamente en escala y posición dando un total de 810000 imágenes, las mismas fueron luego escaladas a un tamaño de 64x128 píxeles (Figura 4)

[1] El tamaño de la imagen depende de la implementación, en esta sección se utilizará a modo de ejemplo 64x128 píxeles.

Figura 3. Imágenes positivas escaladas a una dimensión de 64x128 píxeles

Figura 4. Imágenes negativas escaladas a una dimensión de 64x128 píxeles. Las mismas fueron seleccionadas del conjunto de imágenes del entorno y de una porción del conjunto de imágenes del INRIA.

El conjunto final consta entonces de 810000 *imágenes negativas* y de las 1016 *imágenes positivas*.

Extracción de características de la imagen Un descriptor es usado para transformar una imagen a un espacio de características, generalmente de alta dimensionalidad, que permita resaltar alguna característica de la imagen de forma de poder ser calificada en conjuntos definibles, en este caso árbol y no árbol. El descriptor HOG esta basado en el análisis de contornos mediante el cálculo de gradientes. La idea subyacente que presenta este descriptor es que la apariencia y forma de un objeto en una imagen puede ser representado por la distribución de la orientaciones de los gradientes.

Los pasos para obtener el descriptor (Figura 5) son:

1. Se transforma la imagen a escala de grises.
2. Se calculan los gradientes mediante derivadas parciales en la dirección horizontal y vertical (Ix e Iy). Esto se logra mediante las respectivas convoluciones con la mascara $\begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$ para dx y $\begin{bmatrix} -1 & 0 & 1 \end{bmatrix}^T$ para dy. Luego se calcula dirección y ángulo del gradiente de cada píxel. El resultado es entonces dos matrices, una que represente las magnitudes del gradiente en cada píxel de la imagen original y otra matriz que contenga las orientaciones del mismo.

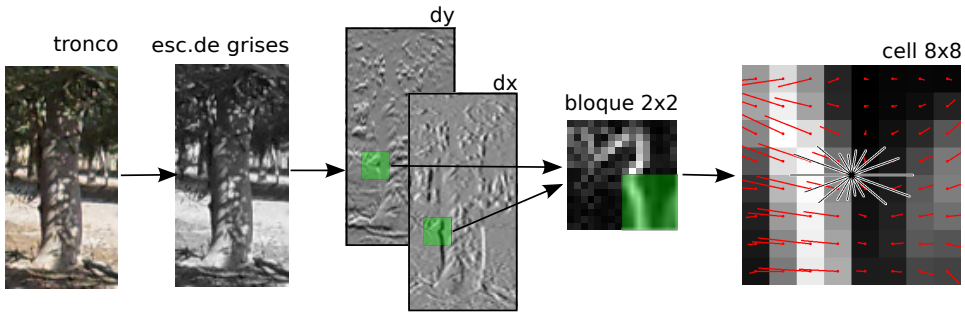

Figura 5. Pasos para obtener el descriptor.

3. Se divide la imagen en bloques descriptores. Estos bloques de un tamaño prefijado están dispuestos de forma tal que cubran toda la imagen a describir con una superposición predefinida entre ellos.
4. Se subdivide cada bloque descriptor en celdas.
5. Por cada celda se calcula un vector, donde cada elemento del mismo posee la suma de las contribuciones de los gradientes para una dirección dada o bins. La cantidad de bins o elementos de ese vector puede variar. Además los bins pueden estar orientados en los 4 cuadrantes si se considera para la contribución de los gradientes magnitud y sentido o solo 2 cuadrantes, si solo se considera magnitud y dirección.
6. Una Normalización de los vectores es usualmente requerida debido a variaciones en el contraste e iluminación.
7. Los vectores de bins de las celdas que forman un bloque son concatenados entre si, resultando en un vector descriptor del bloque.
8. Se concatenan los vectores de cada bloque en un único vector descriptor de la imagen.

2.2. Entrenamiento del clasificador

Se construye un modelo a partir de ejemplos positivos y negativos, y mediante un algoritmo de aprendizaje se aprenden los rasgos mas representativos de cada conjunto de entrenamiento. Con este fin, se utiliza la técnica de clasificación basada en límites de decisión de vectores de apoyo (SVM, Support Vector Machine).

Las SVM se basa en encontrar una superficie de clasificación determinada por ciertos puntos de un conjunto de entrenamiento. Este conjunto de vectores debe estar en la frontera entre los dos subconjuntos en que se clasificarán los puntos, estos vectores son llamados vectores de soporte.

Si se tiene un conjunto de N muestras con vectores xi , etiquetadas en dos clases ($yi = +1$,$yi = -1$) en un espacio de entrada n dimensional (Figura 6), y se necesita determinar cuando una muestra x pertenece a alguna de las dos clases, el objetivo será encontrar (al menos) un hiperplano que proporcione la separación entre las clases[6].

$$\pi : w.x + b = 0$$

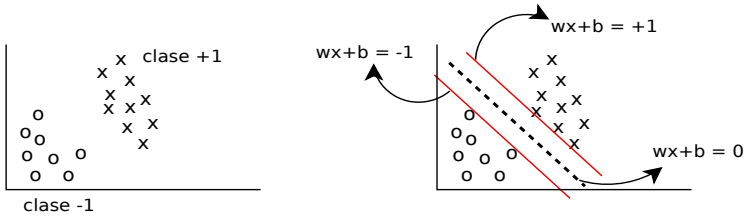

Figura 6. Conjunto de vectores e hiperplano que los separa (en este caso lineal)

La librería de programación de SVM utilizadas fueron LIBLINEAR [4, 1]
El proceso para obtener el clasificador árbol/no-árbol es:

1. Obtener los HOGs de las imágenes positivas.
2. Obtener los HOGs de las imágenes negativas.
3. Entrenar un modelo mediante las SVM lineal para separar los HOGs de las imágenes positivas y negativas.
4. Escaneamos con el clasificador obtenido las imágenes negativas originales, en busca de falsas aceptaciones.
5. Se vuelve a entrenar un modelo en cuyo conjunto de entrenamiento se agregan las falsas aceptaciones obtenidas en el punto anterior como un conjunto adicional de imágenes negativas. De esta forma, obtenemos un modelo mas robusto (*hard model*)

2.3. Fase de detección

Para la fase de detección, se debe realizar el proceso antes descripto, es decir, generar una representación de la imagen en multi-escala y luego, por cada escala, dividir la imagen en bloques del tamaño del clasificador y correr el mismo con el modelo robusto por cada uno de ellos, dando como salida la existencia o no de un árbol en el bloque dado (Figura 7).

Inconvenientes en la aplicación de este método de clasificación Las imágenes utilizadas en el entrenamiento de la SVM y las utilizadas en la fase de detección, deben tener el mismo tamaño, de esta forma, el vector generado en todos los casos posee la misma dimensión. Para un tamaño de imagen mayor, se debe generar una grilla que divida la misma en bloques del tamaño del clasificador, para luego clasificar cada uno de estos bloques de manera aislada.

Otro problema surge con respecto a las escalas. La búsqueda en una grilla de bloques de un tamaño determinado, solo podrá detectar árboles de las dimensiones con que fue entrenado. Se crea entonces una representación de la imagen

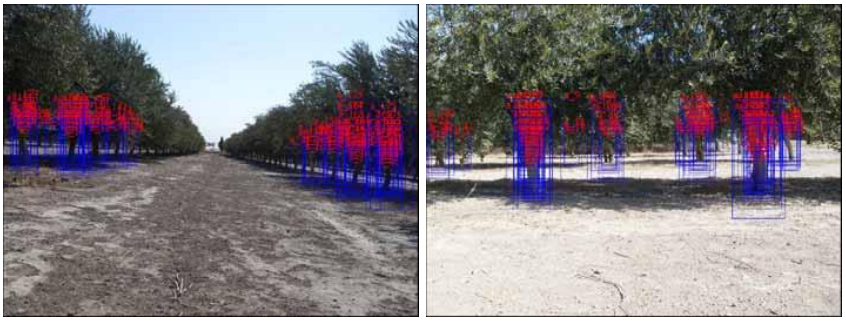

Figura 7. Ejemplos de detecciones de troncos mediante la SVM propuesta en este trabajo. Los rectángulos en azul encierran la porción de la imagen en que fue detectado el tronco y las diferentes escalas en las que sucedió dicha detección. En rojo, son las etiquetas asignadas a cada tronco.

en multi-escala. Esto es, generar por medio de sucesivos filtros de suavizado y posterior downsampling una pirámide de imágenes, donde cada nivel de esta pirámide representa una escala de la imagen original. De esta forma, se detectan imágenes de árboles en distintas escalas.

3. Determinación del tamaño óptimo de la ventana.

Por cada tamaño a evaluar se procede de la siguiente forma:

- Es creado un nuevo grupo de imágenes de entrenamiento y verificación con el tamaño de la imagen a evaluar y son calculados los descriptores de cada imagen como se detalla en 2.1.
- Luego son extraídos un 20 % de imágenes positivas y un 20 % de imágenes negativas de cada conjunto, este nuevo grupo de imágenes y descriptores denominado conjunto de verificación, es utilizado para evaluar la robustez del descriptor entrenado.
- El conjunto resultante luego de la extracción o conjunto de entrenamiento, es utilizado en la etapa de entrenamiento del clasificador como se explica en 2.2
- Finalmente el descriptor obtenido es evaluado con el conjunto de verificación para obtener la robustez del mismo.

El resto de los parámetros fueron los descriptos en el trabajo de Dalal Auat Cheein [2].

- Tamaño de bloque: 2x2 celdas
- Superposición entre bloques: 8 bits
- Cantidad de bins: (9 bins, además se toma la dirección y no el sentido del gradiente, disponiendo a los 9 bins en 180 grados)
- Tamaño de la imagen a describir: Se utilizaron 32x64, 32x72, 64x128 y 48x72 pixeles.

4. Evaluación de resultados

Se ha podido observar una buena robustez del sistema para el clasificador con una ventana de 64x128 píxeles. Con este tamaño del descriptor, se ha logrado detectar el 76 % de los árboles en imágenes y un 93 % de detección de imágenes sin árboles. La comparación para los diferentes tamaños de clasificadores es: Cuadro 1

Tamaño de ventana [pixeles]	Detección de Árboles	Detección de no Árboles
64x128	76 %	93.2 %
32x72	64 %	92.3 %
32x64	59 %	91.8 %
48x72	68 %	92.7 %

Cuadro 1. Comparación de los valores obtenidos en la detección para diferentes tamaño de ventana

A efectos de cuantificar el rendimiento de los diferentes clasificadores se utiliza la curva DET (por su sigla en ingles Detection Error Tradeoff)

4.1. Curva Compensación del error de detección (DET)

Una curva DET muestra el error de falsos negativos contra el grado de error de los falsos positivos, variando el umbral de alguna medida de confianza. La curva nos permite establecer el punto óptimo en el que configurar esta medida, minimizando de este modo el error del sistema.

El análisis se basa en determinar las probabilidades de estos dos tipos de error. Es decir, se grafica la tasa de errores en ambos ejes, dando por igual a ambos tipos de errores.

Las dos acciones posibles de nuestro sistema de detección son: A para aceptar y R para rechazar.

La hipótesis $H = \{ImagenDeArbol\}$ y la negación de esta $\neg H$ conforman un espacio de eventos dado por:

$$\Omega = \{(A, H), (R, H), (A, \neg H), (R, \neg H)\}$$

entonces la probabilidad de error es

$$p(error) = p(R, H) + p(A, \neg H)$$

La curva DET surge de graficar $P(R, H)$ respecto de $P(A, \neg H)$

$$P(A, \neg H) = FPPW$$

$$P(R,H) = 1 - \frac{\#DeteccionesVerdaderasPositivas}{\#TotalDePositivas} = MissRate$$

o también

$$MissRate = \frac{\#FalsasNegativas}{\#FalsasNegativas + \#VerderasPositivas}$$

La gráfica (Figura 8) muestra los resultados obtenidos para diferentes tamaños del clasificador, en el eje de abscisa FPPW y en el eje de ordenadas el miss rate. Valores bajos indican un mejor rendimiento del clasificador.

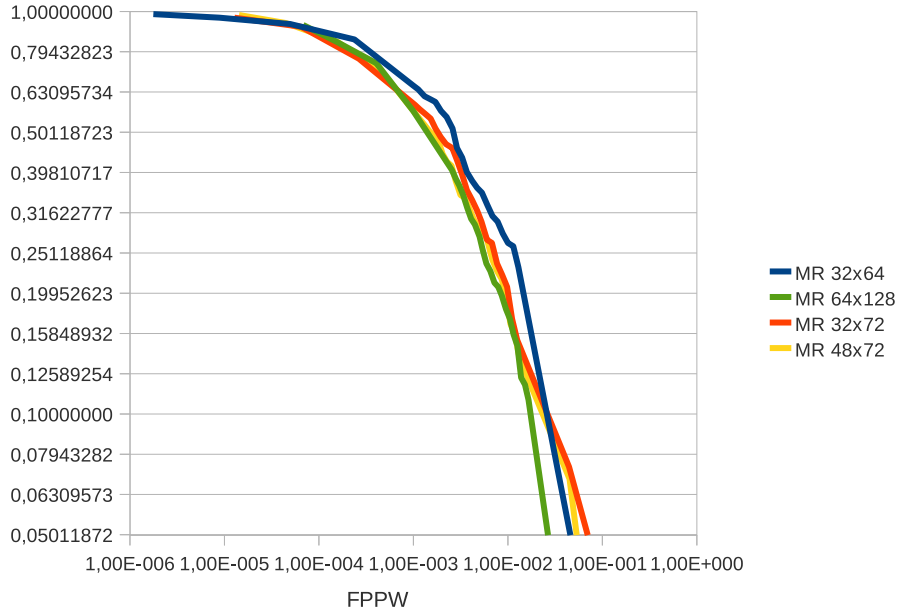

Figura 8. Resultados obtenidos para diferentes tamaños del clasificador

Para todos los detectores el método parte de la puntuación más baja posible, se evalúa los respectivos parámetros como el número de falsos positivos, el índice de repetición o la tasa de precisión, y luego progresivamente se aumenta el umbral hasta que lleguen a la mayor puntuación posible. Cada umbral de prueba proporciona un punto de la curva.

5. Conclusiones

En este trabajo se presentó una implementación de un sistema de detección para la extracción de características extraídas de un entorno agrícola correspondiente a troncos asociados a las plantas de olivos del entorno. Para obtener el método de detección se utilizaron tres etapas. En una primera etapa se preparó el conjunto de imágenes con los que se iba a trabajar. Una segunda etapa, basada en obtener descriptores HOGs de las imágenes y una tercera etapa, donde se utilizan máquinas de soporte vectorial para crear un clasificador. De esta forma se obtuvo un modelo, el cual permite detectar los troncos en el plano de una imagen en diferentes posiciones y escalas.

Los resultados experimentales han demostrado la efectividad de utilizar descriptores HOGs para la detección.

Bibliografía

[1] Machine Learning Group at National Taiwan University. Liblinear – a library for large linear classification. URL http://www.csie.ntu.edu.tw/ cjlin/liblinear/.

[2] G. Perez Paina Carelli Auat Cheein, Guillermo Steiner. Aplicación de un eif-slam en entornos agrícolas basado en detección de troncos de árboles.

[3] Mark Everingham, Andrew Zisserman, Christopher K. I. Williams, Luc Van Gool, Moray Allan, Christopher M. Bishop, Olivier Chapelle, Navneet Dalal, Thomas Deselaers, Gyuri Dorkó, Stefan Duffner, Jan Eichhorn, Jason D. R. Farquhar, Mario Fritz, Christophe Garcia, Tom Griffiths, Frederic Jurie, Daniel Keysers, Markus Koskela, Jorma Laaksonen, Diane Larlus, Bastian Leibe, Hongying Meng, Hermann Ney, Bernt Schiele, Cordelia Schmid, Edgar Seemann, John Shawe-taylor, Amos Storkey, Or Szedmak, Bill Triggs, Ilkay Ulusoy, Ville Viitaniemi, and Jianguo Zhang. The 2005 pascal visual object classes challenge. 2006.

[4] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, pages 1871–1874, 2008.

[5] INRIA. Inria person data set. URL http://pascal.inrialpes.fr/data/human/.

[6] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf. An introduction to kernel-based learning algorithms. *IEEE Neural Networks*, 12(2): 181–201, May 2001.