DETECCION DE OBJETOS BASADOS EN HOG/SVM EN UNA IMAGEN

Jonatan Villa Morales
Universidad Autónoma de Guerrero.
Chilpancingo, Gro., México
+51 1 7472752401
Jvm11mng@gmail.com

Mario Hernández Hernández
Universidad Autónoma de Guerrero.
Chilpancingo, Gro., México
+52 1 7471120661
mhernandezh@uagro.mx

Severino Feliciano Morales
Universidad Autónoma de Guerrero.
Chilpancingo, Gro., México
+52 1 7474798207
sevefelici72@gmail.com

José Luis Hernández Hernández
Universidad Autónoma de Guerrero.
Chilpancingo, Gro., México
+52 1 7471618004
jlhernandez@uagro.mx

RESUMEN

En este trabajo de investigación cuando hablamos de detección de objetos, nos referimos a utilizar técnicas de visión por computadora. Podemos pensar en tener una sola imagen y detectar objetos dentro de esa imagen, o podemos pensar en tener vídeos, y por tanto una secuencia de imágenes, y detectar objetos en esa secuencia, incluso podemos pensar en tener varias imágenes, y por tanto que podemos usar no solo la apariencia de los objetos sino también la distancia a la cámara y su tamaño.

Palabras Clave

Tecnología, Visión por computadora, Detección de objetos, Imagen, Hog-Svm.

ABSTRACT

In this research work when we talk about object detection, we refer to using computer vision techniques. We can think of having a single image and detect objects within that image, or we can think of having videos, and therefore a sequence of images, and detect objects in that sequence, we can even think of having several images, and therefore we can use not only the appearance of the objects but also the distance to the camera and their size.

Keywords

Technology, Computer vision, Object detection, image, Hog-Svm

INTRODUCCIÓN

La visión artificial o visión por computadora es una disciplina compleja que involucra otras ciencias e incluye estudios de física, matemáticas, ingeniería electrónica e ingeniería informática entre otras. El continuo desarrollo de algoritmos, funciones y aplicaciones hace que sea una disciplina en continua evolución. La visión artificial es un subcampo de la inteligencia artificial y su propósito es programar una computadora para que "entienda" una escena o las características de una imagen.

Como es evidente, el diseño de un sistema de visión artificial por computadora intenta simular lo que una persona humana capta con el sentido de la vista. Es decir: reconocimiento de figuras, objetos, distancia hasta ellos, textura que lo conforma y todas las características que un humano deduce de un objeto con solo verlo.

Los primeros conocimientos que se tienen de esta materia se remontan a los años veinte del siglo pasado, cuando se mejora la calidad de las imágenes digitalizadas de los periódicos, enviadas por cable submarino entre Londres y Nueva York. Sin embargo, no es hasta la década de los 50's cuando empiezan a aparecer los primeros trabajos relacionados con la visión artificial. Al principio se piensa que es una tarea sencilla y alcanzable en pocos años, esto se debe a los importantes trabajos realizados por Roberts en 1963 [5] y Wichman en 1968 [6]. El primero demuestra la posibilidad de procesar una imagen digitalizada para obtener una descripción matemática de los objetos que aparecían y el segundo presenta por primera vez una cámara de televisión conectada a una computadora.

En la década de los ochenta vuelven a aparecer las investigaciones relacionadas con la visión por computadora, en este caso encaminadas a la extracción de características. Así se tiene la detección de texturas [7] y la obtención de la forma a través de ellas [8]; y ese mismo año, 1981, se publican artículos sobre: visión estéreo (Mayhew y Frisby [9]), detección de movimiento (Horn [10]) e interpretación de formas (Steven); o los detectores de esquina (Kitechen y Rosendfekd, en 1982 [11]). A pesar de la importancia de las investigaciones y artículos recién comentados, el trabajo más importante de la década es el libro de David Marr [12], donde se abordaba por primera vez una metodología completa del análisis de imágenes a través de computadora.

La visión, de una manera simple y resumida, consiste en capturar imágenes y procesar el contenido que hay en ellas para obtener información. Para una computadora, la parte de captación de imágenes ya está hecha. Tan solo debemos utilizar el hardware adecuado para capturar imágenes (cámaras web, cámaras digitales, videocámaras, etc.) y, una vez obtenidas estas imágenes, debemos realizar la parte de procesamiento de imágenes, aunque esta fase es una ardua tarea. Con el procesamiento de imágenes, se puede establecer la relación entre el mundo tridimensional y las vistas bidimensionales tomadas de él. Se puede hacer, por una parte, una reconstrucción del espacio tridimensional a partir de sus vistas y, por otra parte, llevar a cabo

una simulación de una proyección de una escena tridimensional en la posición deseada a un plano bidimensional.

Uno de los principales objetivos de la visión por computadora es poder diferenciar los objetos presentes en una imagen (en este caso, en una imagen digital), de tal manera, que esto logre que la identificación de estos (un proceso posterior) sea una tarea más fácil de realizar. Cuando nos enfocamos en el problema de diferenciar los objetos en cualquier imagen (como es hecho por los seres humanos), una importante pregunta abierta aparece: ¿Qué información es, tanto suficiente como necesaria, para poder llevar a cabo esta tarea? Es muy difícil poder expresar este conocimiento o información de una manera algorítmica, por lo que esta respuesta es contestada solamente para algunos casos particulares y no en general. Dentro de la Visión Computacional, podemos encontrar el procesamiento de imágenes, y dentro de este, una parte muy importante se encarga del análisis de estas. Esto es, dada una imagen, lo que deseamos obtener es una descripción de dicha imagen. Los siguientes son ejemplos de problemas de análisis de imágenes: 1) Dado un texto, reconocer las palabras. 2) Dada una imagen aérea de un terreno, clasificar los distintos tipos de suelos (urbano, bosque, lagos, carreteras, etc.). 3) Dada una imagen de un conjunto de células, clasificar las células por tamaño, forma, etc. 4) Dada una imagen médica, detectar tumores, roturas de huesos, etc. Es decir, dada una imagen, el análisis se encarga de entregar información de ella. Por lo que, en todos estos ejemplos, el análisis depende primeramente de detectar determinadas partes de la imagen (regiones u objetos). Para generar tal descripción es necesario segmentar (o separar) adecuadamente e identificar la región deseada.

Existen diversas técnicas para el análisis de formas en imágenes. Algunos autores utilizan técnicas basadas en extracción de características y patrones de formas [1], [2], técnicas de Boosting, detección de objetos basada en formas mediante métodos Chamfer, correlación con patrones humanos probabilísticos, máquinas de soporte vectorial (SVM) [3], graph kernels, análisis de movimiento, análisis de componentes principales y clasificadores basados en redes neuronales.

El análisis de imágenes es utilizado en detección de caras y reconocimiento facial. Es también utilizado en seguimiento de objetos, por ejemplo, siguiendo una pelota durante un partido de fútbol o siguiendo una persona en un video.

CAPITULO 1 (MARCO TEORICO)

La imagen digital será la entrada de nuestro sistema detector de objetos y así pues vamos a ver como se representa esta y los elementos que intervienen en su formación. Este sistema tendrá siempre una entrada y una salida. La entrada será una imagen y un indicador del objeto que queremos detectar. Y la salida será la ventana o conjunto de ventanas que hemos detectado con el objeto indicado.

El color que requiere un píxel depende de 3 componentes, tal como se muestra en la Figura 1.

- El color de la luz
- El material de la superficie
- La sensibilidad de la cámara

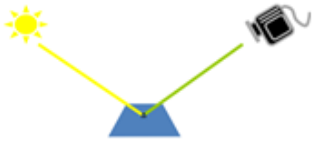


Figura 1. Componentes que requiere el color de un pixel

El color es una característica de la luz que puede ser observada por los humanos, y depende de la longitud de onda, tal como se muestra en la Figuras 2, 3 y 4.

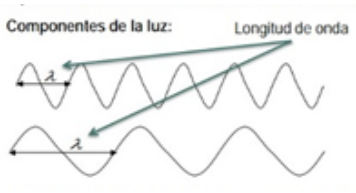


Figura 2. El color

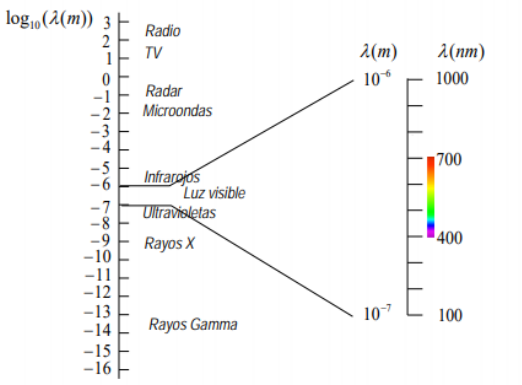


Figura. 3 Luz visible

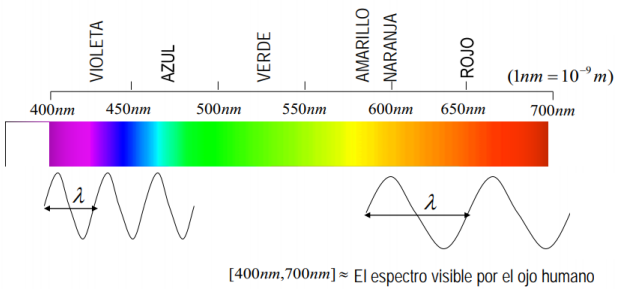


Figura 4. Color y longitud de onda de la luz

El material de la superficie, mostrado en la Figura 5 se aprecia la naturaleza de cada material, por ejemplo, sus pigmentos, determina las longitudes de onda que refleja y las que absorbe.

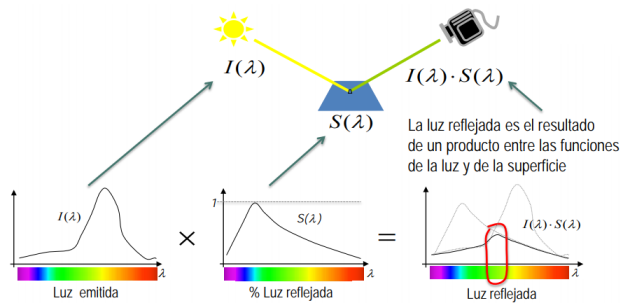


Figura 5. Material de la superficie

En cuanto a la sensibilidad de la cámara, las cámaras tienen tres tipos de sensores que integran sobre diferentes longitudes de onda con el objetivo de cubrir todo el espectro visible, tal como se muestra en la Figura 6.

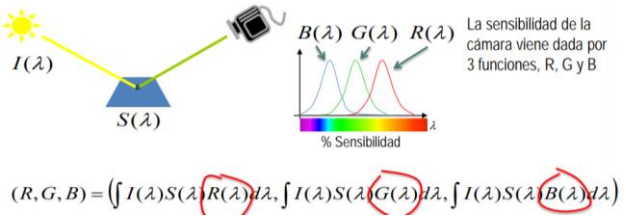


Figura 6. Sensibilidad de la cámara

Otro tipo de imágenes (más allá de la visión humana) son las imágenes de infrarrojos cercanos (RGB-NIR), imágenes térmicas, así como las imágenes de profundidad RGBD.

Las imágenes de infrarrojos cercanos RGB-NIR (Near Infrared), corresponde a imágenes que añaden un canal no visible que engloba las longitudes de onda de los infrarrojos de 700nm hasta 1100nm. Tal como se muestra en la figura 7.

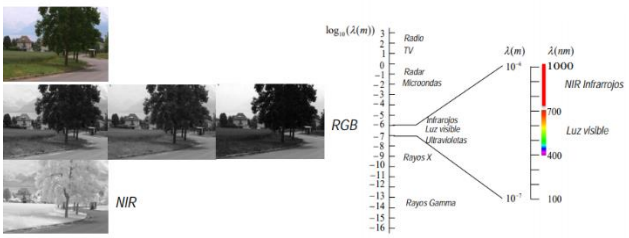


Figura 7. Imagen infrarrojos cercanos RGB-NIR

Las imágenes térmicas (Long-Wavelength Infrared), se refiere a las imágenes térmicas que reproducen la correlación existente entre la temperatura y la emisión infrarroja de los objetos. Esta emisión infrarroja se encuentra en el intervalo de longitudes de onda que va de 8µm a 15µm (infrarrojo de onda larga), tal como se muestra en la Figura 8.

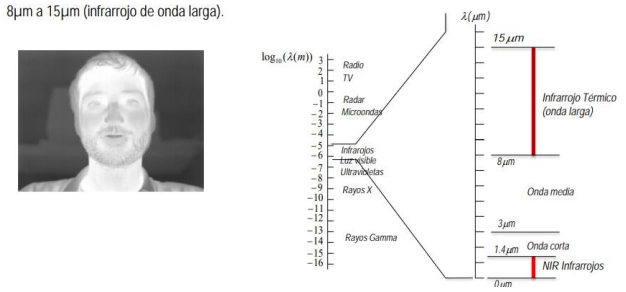


Figura 8. Imágen térmica (Long-Wavelength Infrared)

Las imágenes de profundidad RGBD (D: Depth), son imágenes capturadas con dispositivos específicos que añaden un canal en el que se estima un mapa de profundidad de la escena que codifica la distancia de cada punto de una superficie con el sensor. Ver la Figura 9.

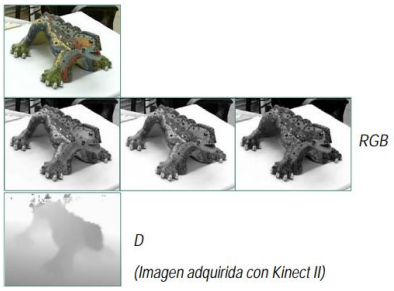
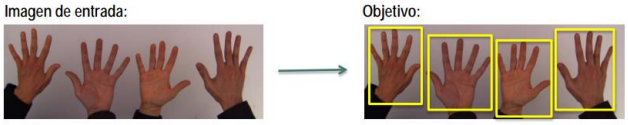


Figura 9. Imagen de profundidad RGBD

Características del píxel

La extracción de características constituye los descriptores de los objetos de las imágenes. Ver Figuras 10 y 11.

Figura 10. Descriptores de las imágenes



Objeto: Región de píxeles conectados que tienen color de piel.

Figura 11. Descriptor simple: El color del pixel

Localización de objetos en imágenes

Cuando queremos localizar un objeto en una imagen, una técnica utilizada consiste en recorrer toda la imagen aplicando múltiples ventanas detectoras de diferentes tamaños. Para cada ventana se aplica algún clasificador que nos indica si existe el objeto que

intentamos. Al ser un algoritmo que se basa en diversos recorridos de la imagen, el desplazamiento de la ventana a través de ella es un factor muy importante. Con la disminución del desplazamiento de la ventana, el tiempo de proceso crece exponencialmente. Por otro lado, un desplazamiento demasiado grande podría comportar la no detección de objetos por no quedar localizados dentro de su ventana.

Objetos y fondos

El separar la imagen en unidades significativas es un paso importante en visión computacional para llegar al reconocimiento de objetos. Este proceso se conoce como segmentación. Una forma de segmentar la imagen es mediante la determinación de los bordes. El dual de este problema es determinar las regiones; es decir, las partes o segmentos que se puedan considerar como unidades significativas. Esto ayuda a obtener una versión más compacta de la información de bajo nivel, ya que, en vez de miles o millones de pixeles, se puede llegar a decenas de regiones, y de ahí reconocer los objetos.

Las características más comunes para delimitar o segmentar regiones son: intensidad de los píxeles, textura, color y gradiente. Una suposición importante, que normalmente se asume en visión de nivel intermedio, es considerar que píxeles de un mismo objeto comparten propiedades similares. Por ejemplo, al procesar una imagen de una manzana, suponemos que el color de sus píxeles es aproximadamente homogéneo. En la vida real esto no es totalmente cierto, el color de los píxeles varía. Para evitar este tipo de variaciones es mejor considerar un color "aproximadamente" similar sobre una región más que a nivel píxel. Esto no es un problema sencillo, ya que es difícil distinguir las variaciones propias del objeto o por cambios de iluminación (por ejemplo, cuando hay sombras en los objetos), de las diferencias por tratarse de otro objeto.

En el análisis de los objetos en imágenes es esencial que podamos distinguir entre los objetos del interés y "el resto", normalmente se hace referencia a este último grupo como el fondo. Las técnicas que se utilizan para encontrar los objetos de interés se refieren generalmente como técnicas de segmentación (segmentar el primer plano del fondo). El resultado de esta segmentación da como resultado una imagen binaria. Normalmente se utiliza la convención de que se le asignan el valor 1 a los píxeles que corresponden a objetos y el valor de 0 a los píxeles que corresponden al fondo. Como resultado de dicha segmentación, la imagen es partida en regiones y se conocen los bordes entre las regiones.

CAPITULO 2: DETECCION DE OBJETOS

La detección de objetos y el cálculo de la profundidad a la que se encuentran son dos procesos de gran utilidad en multitud de ámbitos, como la robótica, la supervisión y el control de calidad, la ayuda a personas con alguna discapacidad o la conducción automática, por enumerar sólo unos pocos. La visión estereoscópica puede ser una solución, aunque hoy en día es todavía un campo abierto de investigación. Se han conseguido resultados adecuados en entornos simples, sin embargo, el cálculo de la profundidad en determinados casos es un problema muy complejo, especialmente cuando hay poca textura, existen oclusiones, etc. Por otro lado, los procesos asociados a la visión estereoscópica (emparejamiento, cálculo de disparidades, etc.) son procedimientos intrínsecamente complejos, con costos temporales muy altos. Una posible aproximación al problema es la segmentación previa de las imágenes estereoscópicas. Estas técnicas dividen una o, a veces, las dos imágenes estéreo en regiones no solapadas de color homogéneo. En lugar de calcular la disparidad para cada píxel individual, estas técnicas asignan un único valor de disparidad a cada una de las regiones obtenidas. Este planteamiento presenta dos ventajas: por un lado, el hecho de utilizar regiones en vez de píxeles hace el proceso más robusto frente a la presencia o no de texturas; por otro lado, la cantidad de emparejamientos a realizar entre las regiones es mucho menor que el que se tendría que realizar para el total de píxeles. Sin embargo, la reducción del número de emparejamientos tiene una contrapartida: debemos realizar un proceso previo de segmentación, cuestión no trivial, debemos elegir un conjunto de propiedades que caractericen convenientemente a las regiones para poder realizar la correspondencia.

Hay una considerable cantidad de literatura sobre el problema de la correspondencia en estéreo. En él se puede encontrar una extensa revisión de los algoritmos actuales de estéreo que producen un mapa de disparidad denso (con información para todos los puntos de la imagen). Centrándonos en técnicas que emplean la visión artificial o visión por computadora se encarga de la obtención de datos de una cámara, su procesamiento y análisis con el fin de producir información que pueda ser tratada por una computadora.

La detección de objetos es una tecnología de computadora relacionada con la visión artificial y el procesamiento de imagen que trata de detectar casos de objetos semánticos de una cierta clase (como humanos, edificios, o coches) en vídeos e imágenes digitales. Los ámbitos mejor desarrollados de detección de objetos incluyen detección de caras y detección de personas. La detección de objetos tiene aplicaciones en muchas áreas de visión artificial, incluyendo recuperación de imágenes y videovigilancia.

Técnicas para la detección

Una imagen está hecha de píxeles, de modo que en la mayoría de los casos sabemos la ubicación del próximo punto, que estará junto a nuestro píxel actual.

Para identificar círculos, la imagen se transforma a escala de grises y se detectan los contornos. A lo largo de los contornos, se trazan las normales, que interceptarán en el centro. Esto sirve para círculos enteros. Otro algoritmo consiste en que a lo largo de los contornos conectados la rotación de la tangente será uniforme, debido a simetría. De este modo, si hay un cambio repentino en la rotación, está fuera del círculo.

Para identificar cuadrados, ante todo hay que comprobar si son líneas rectas o no (comprobar si los píxeles tienen las mismas coordenadas x o y). Después, buscar un cambio con un ángulo de 90 grados (si se mueve a lo largo de una línea horizontal, en la esquina la coordenada y parará cambiar y la coordenada x empezará cambiar). La detección de personas en secuencias reales resulta ser un reto debido a las variadas formas en las que se

pueden encontrar las personas. Ahora trataremos diferentes técnicas para la detección y clasificación.

Técnicas de extracción de características

Es el proceso de extraer características que puedan ser usadas en el proceso de clasificación de los datos. En ocasiones viene precedido por un preprocesado de la señal, necesario para corregir posibles deficiencias en los datos debido a errores del sensor, o bien para preparar los datos de cara a posteriores procesos en las etapas de extracción de características o clasificación.

Las características elementales están explícitamente presentes en los datos adquiridos y pueden ser pasados directamente a la etapa de clasificación. Las características de alto orden son derivadas de las elementales y son generadas por manipulaciones o transformaciones en los datos.

A continuación, detallamos brevemente el funcionamiento de los más importantes:

Descriptores wavelet de haar

Este método fue propuesto por Viola y Jones en 2004 [13]. Los descriptores Wavelet de Haar permiten definir de manera robusta clases de objetos complejos, siendo invariantes a cambios de color y de textura. Se emplean habitualmente para la descripción de personas. Presentan la capacidad de codificar rasgos tales como cambios de intensidades a diferentes escalas. La base de wavelet más sencilla es la de Haar que consiste en que se recorre la imagen con una ventana a la que se le aplican varios clasificadores en serie, cada uno más complejo que el anterior, los cuales usan las características para confirmar o descartar la hipótesis de que se trata del objeto buscado. Si la hipótesis se rechaza en cualquier nivel, el proceso no continúa, pero si se confirma todos los filtros significará que se ha detectado el objeto deseado. Los patrones se consideran girados en varios posibles ángulos. Además, el algoritmo puede ejecutarse a varias escalas para obtener objetos de diferentes tamaños o de tamaño desconocido.

SIFT (Scale-Invariant Feature Transform)

SIFT es un método propuesto por David Lowe en 1999 [14], que se centra en buscar puntos característicos que cumplen criterios espacio-escalares. Los descriptores se calculan a través de la orientación de los gradientes de cada punto. Así se extraen puntos característicos invariantes y distintivos de una imagen que pueden ser usados para mejorar la correspondencia entre dos vistas diferentes de un objeto o una escena.

SURF (Speeded Up Robust Feature)

Es uno de los sucesores más importantes de SIFT, ha sido el algoritmo Speeded-Up Robust Features (SURF) [15]. SURF fue presentado en 2006 en el ECCV en Graz (Austria). Está parcialmente inspirado en SIFT y se ha demostrado que en la práctica, la totalidad de los casos consigue mejorar el rendimiento de este algoritmo [16]. Se basa en el cálculo del determinante de la matriz Hessiana (DoH: Determinant of Hessian) para la detección de puntos interesantes y en las wavelets de Haar para la descripción de dichos puntos. Esta aproximación es aún más rápida que DoG (la utilizada por SIFT) y ofrece una respuesta superior en cuanto a calidad de descripción de las imágenes.

Detectores de bordes canny

Fue desarrollado por John F. Canny en 1986, utiliza un algoritmo de múltiples etapas para detectar una amplia gama de bordes en imágenes [17]. Extraen los bordes de los objetos en las imágenes mediante la selección de aquellas regiones con altas derivadas espaciales. El hecho de tener en cuenta sólo los bordes de los elementos en la imagen reducen significativamente el tamaño de los datos a tratar, y filtra la información no útil de la imagen, conservando las formas, que es lo que proporciona la información relevante.

HOG

Este método fue presentado por Navneed Dalal y Bill Triggs en el Instituto Nacional de Investigación en Informática y Automática (INRIA), en 2005 [2]. Consiste en la división de la imagen en subbloques distribuidos a lo largo y ancho de la misma y con cierto solape entre ellos. Cada bloque se subdivide en subloques (o celdas) y sobre estos últimos se calcula la magnitud y orientación de los gradientes en cada píxel. Sobre cada uno de estos bloques se calcula el histograma de los gradientes orientados promediado por un peso gausiano, y luego se almacena en el vector de características de la imagen.

CAPITULO 3: CASO PRACTICO: DETECTOR BASADO EN HOG/SVM EN UNA IMAGEN

Los descriptores HOG (del inglés Histogram of Oriented Gradients-HOG) se basan en la orientación del gradiente en áreas locales de una imagen. La imagen se divide en pequeñas celdas cada una de las cuales acumula direcciones del histograma de gradiente u orientaciones de los bordes de los píxeles de las celdas. Se recomienda para una mejor respuesta normalizar el contraste en unas zonas más grandes (denominadas bloques) y utilizar dicho resultado para normalizar las celdas del bloque. Estos bloques de descriptores normalizados son lo que los autores denominan descriptores HOG (Figura 12). Por último, se utilizan los descriptores HOG de la ventana de detección como entrada a un clasificador SVM.

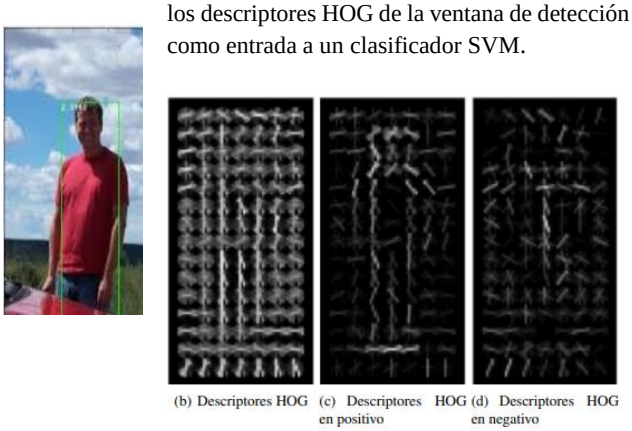


Figura 12. Ejemplo de la extracción de descriptores HOG

Los descriptores HOG nos proporciona información tal como los cambios de intensidad debido a los contornos o bordes de una imagen. Al tener en cuenta la relación con sus zonas vecinas y colindantes, es posible reconocer cuándo existe una frontera entre un objeto y otro. De esta manera, podremos identificar objetos de siluetas más suaves o más pronunciadas. El descriptor de HOG es, por lo tanto, especialmente adecuado para la detección de personas, independientemente de su tamaño y sus colores, y fijándonos más en su relación con el entorno, distinguiendo los cambios más pronunciados.

Las bases teóricas de los métodos HOG residen en trabajos previos tales como Histogramas de bordes orientados (Freeman and Roth 1995 [18]), descriptores SIFT (Lowe 1999 [14]) y reconocimiento de formas (Belongie et al. 2001 [19]), entre otros. Sin embargo, la diferencia añadida que presentan los métodos HOG consiste en que los gradientes no se calculan uniformemente sobre un mallado denso, sino que se divide la imagen en bloques y, a su vez, cada bloque en diversos sub-bloques, y se calcula en cada uno de ellos los gradientes y el histograma.

El algoritmo HOG es capaz detectar la presencia de peatones presentes en una escena. Una vez detectados aquellos peatones, que por su cercanía al robot corren peligro de ser atropellados, el robot puede ser alertado de la presencia de éstos con la suficiente antelación como para poder reaccionar en el caso de que exista un riesgo. Es por ello que se instala que el programa procese las imágenes tomadas con la mayor brevedad posible. El cálculo de los descriptores HOG no presenta un costo de tiempo de computación bastante elevado por el hecho de calcular el HOG en cada una de las celdas. La elección de este método para llevar a cabo la detección de los peatones en la escena se basa en que destaca por su robustez frente a diferentes condiciones de iluminación, pequeños cambios en el contorno de la imagen y diferencias de fondos y de escalas. Los descriptores propuestos se basan en trabajos previos, tales como histogramas de bordes orientados, descriptores SIFT y reconocimiento de formas. Dada una imagen en color, lo primero que se hace es transformar a escala de grises. A continuación, se calculan los gradientes espaciales sobre toda la imagen. Posteriormente, se divide la imagen en bloques, solapados cierta área.

El avance de bloques se realiza eliminando la columna de las celdas de la izquierda y añadiendo la columna de la derecha para el desplazamiento horizontal, mientras que para el vertical se elimina la fila de las celdas de arriba, añadiendo la fila de celdas de abajo. A su vez, cada bloque se divide en subregiones o celdas, calculándose en cada uno de ellos el histograma de los gradientes orientados, de tal forma que se logra mejorar el rendimiento.

Finalmente se aplica una ventana gausiana sobre cada bloque, almacenándose dicha información en el vector de características de la imagen.

HOG (Histograma De Gradientes Orientados)

HOG significa Histogramas de Gradientes Orientados. HOG es un tipo de "descriptor de características". El objetivo de un descriptor de características es generalizar el objeto de tal forma que el mismo objeto (en este caso un rostro produzca lo más cerca posible del mismo descriptor de características cuando se lo vea bajo diferentes condiciones. Esto hace que la tarea de clasificación sea más fácil.

Los creadores de este enfoque formaron un Support Vector Machine (un tipo de algoritmo de aprendizaje automático para la clasificación), o "SVM", para reconocer los descriptores HOG de los rostros.

El detector de rostros HOG es bastante simple de entender (en comparación con el reconocimiento de objetos SIFT, por ejemplo). Una de las principales razones para esto es que utiliza una función "global" para describir un rostro en lugar de una colección de características "locales". En pocas palabras, esto significa que todo rostro está representado por un único vector de características, a diferencia de muchos vectores de características que representan partes más pequeñas de ese rostro.

El detector de rostros HOG usa una ventana de detección deslizante que se mueve alrededor de la imagen. En cada posición de la ventana del detector, se calcula un descriptor HOG para la ventana de detección. Este descriptor se muestra luego al SVM entrenado, que lo clasifica como "rostro" o "no rostro".

Para reconocer los rostros a diferentes escalas, la imagen se submuestrea en varios tamaños. Se busca cada una de estas imágenes submuestreadas

Máquinas de Soporte Vectorial (SVM)

Las Máquinas de Soporte Vectorial [4] son estructuras de aprendizaje basadas en la teoría estadística del aprendizaje. Se basan en transformar el espacio de entrada en otro de dimensión superior (infinita) en el que el problema puede ser resuelto mediante un hiperplano óptimo (de máximo margen).

Estos métodos están propiamente relacionados con problemas de clasificación y regresión. Dado un conjunto de ejemplos de entrenamiento (de muestras) podemos etiquetar las clases y entrenar una SVM para construir un modelo que prediga la clase de una nueva muestra. Intuitivamente, una SVM es un modelo que representa a los puntos de muestra en el espacio, separando las clases por un espacio lo más amplio posible. Cuando las nuevas muestras se ponen en correspondencia con dicho modelo, en función de su proximidad pueden ser clasificadas como pertenecientes a una u otra clase.

Más formalmente, una SVM construye un hiperplano o conjunto de hiperplanos en un espacio de dimensionalidad muy alta (o incluso infinita) que puede ser utilizado en problemas de clasificación o regresión. Una buena separación entre las clases permitirá una clasificación correcta.

Las Máquinas de Soporte Vectorial son estructuras de aprendizaje basadas en la teoría estadística del aprendizaje. Se basan en transformar el espacio de entrada en otro de dimensión superior (infinita) en el que el problema puede ser resuelto mediante un hiperplano óptimo (de máximo margen). Estos métodos están propiamente relacionados con problemas de clasificación y regresión. Dado un conjunto de ejemplos de entrenamiento (de muestras) podemos etiquetar las clases y entrenar una SVM para construir un modelo que prediga la clase de una nueva muestra. Intuitivamente, una SVM es un modelo que representa a los puntos de muestra en el espacio, separando las clases por un espacio lo más amplio posible. Cuando las nuevas muestras se ponen en correspondencia con dicho modelo, en función de su proximidad pueden ser clasificadas como pertenecientes a una u otra clase. Más formalmente, una SVM construye un hiperplano o conjunto de hiperplanos en un espacio de dimensionalidad muy alta (o incluso infinita) que puede ser utilizado en problemas de clasificación o regresión. Una buena separación entre las clases permitirá una clasificación correcta. Idea básica Las Máquinas de Soporte Vectorial (SVM) son un conjunto de algoritmos de aprendizaje supervisado empleados para la clasificación y la regresión. Dado un conjunto de ejemplos de entrenamiento (muestras) podemos etiquetar las clases y entrenar una SVM para construir un modelo que prediga la clase de una nueva muestra. Tomando los datos de entrada como conjuntos de vectores en un espacio n-dimensional, una Máquina de Soporte Vectorial construirá un hiperplano de separación en ese espacio (Figura 13). Se considera que es mejor clasificador de datos aquel hiperplano que maximice la distancia (o margen) con los puntos que estén más cerca de él. Siendo los vectores de soporte los puntos que tocan el límite del margen. En el contexto que se está tratando en este proyecto de detección de personas, las clases de datos corresponderán al humano (muestras positivas), mientras que el resto de la imagen será tachada como muestras negativas. La SVM busca un hiperplano que separe de forma óptima a los puntos de una clase de la de otra, que eventualmente han podido ser previamente proyectados a un espacio de dimensionalidad superior.

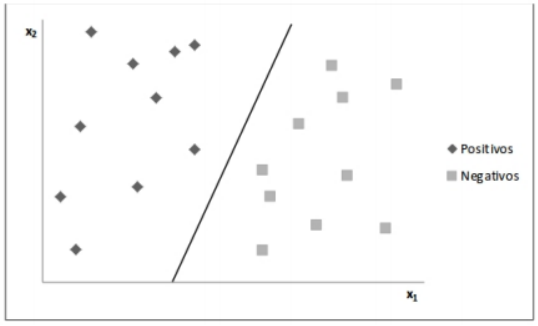


Figura 13. Hiperplano de separación de dos clases.

En ese concepto de "separación óptima" es donde reside la característica fundamental de las SVM: este tipo de algoritmos buscan el hiperplano que tenga la máxima distancia (margen) con los puntos que estén más cerca de él mismo. Por eso, también a veces se les conoce a las SVM como clasificadores de margen máximo. De esta forma, los puntos del vector que son etiquetados con una categoría estarán a un lado del hiperplano y los casos que se encuentren en la otra categoría estarán al otro lado. Para separar linealmente los datos se procede a realizar un cambio de espacio mediante una función que transforme los datos de manera que se puedan separar linealmente. Esta función recibe el nombre de Kernel.

En este caso, los conjuntos son "personas" y "no personas". Para ello, se necesita un entrenamiento previo de la máquina, facilitándole ejemplos de personas o "positivos" y ejemplos de no-personas o "negativos". Con todos los ejemplos de entrenamiento, el algoritmo de clasificación SVM elabora una curva M-dimensional que divide ambos conjuntos, obteniendo de esta forma el kernel de la máquina. Las dimensiones del espacio dependen del número de componentes de cada vector a clasificar.

SVM^{light}

El proceso de clasificación mediante una Máquina de Soporte Vectorial consta de dos pasos: entrenamiento y clasificación, donde en el primero se reconocen los patrones del conjunto de datos de entrenamiento con el fin de crear un modelo que luego será empleado en la clasificación de nuevos datos. Este proceso presenta complejidad de orden cuadrático respecto a las dimensiones de los datos de entrenamiento por lo que los problemas que se pueden solucionar con esta técnica se ven limitados. Actualmente existen tres algoritmos fundamentales [20] para el entrenamiento de SVM en software: Chunking [21], Sequential Minimum Optimization (SMO) [22] y SVM [23]. Este último es una mejora propuesta al algoritmo planteado en el trabajo de Osuna, "Improved Training Algorithm for Support Vector Machines [24] ". El algoritmo tiene requisitos de memoria escalable, resuelve problemas de clasificación y regresión, y, por lo tanto, es el más adecuado para la detección de personas.

SVM^{light} [4] es una implementación de SVM en C, con variantes para aprendizaje supervisado, y para semisupervisado transductivo. Hemos utilizado esta implementación ya que se puede utilizar su código para investigación. Las principales características del programa son los siguientes:

- Algoritmo de optimización rápida.
- Resuelve problemas de clasificación y regresión.
- Calcula XiAlpha de las estimaciones de la tasa de error, precisión y recall.
- Incluye algoritmo para la formación de unos grandes SVMs transductivos (TSVMs).
- Se puede entrenar a SVMs con los modelos de costos y gastos que dependen del ejemplo.
- Maneja hasta diez mil ejemplos de entrenamiento.
- Maneja varios miles de vectores de soporte.
- Soporta funciones de núcleo estándar.
- Usa representación por vector disperso.

Detección de rostros

La detección de rostros se generalizó a principios de la década de 2000, cuando Paul Viola y Michael Jones inventaron una forma de detectar caras lo suficientemente rápido como para funcionar con cámaras baratas. Sin embargo, existen soluciones mucho más confiables ahora. Vamos a utilizar un método inventado en 2005 llamado Histograma de gradientes orientados, o simplemente HOG para abreviar.

Para encontrar rostros en una imagen, empezaremos haciendo que nuestra imagen sea en blanco y negro porque no necesitamos datos de color para encontrar rostros. Ver Figura 14.

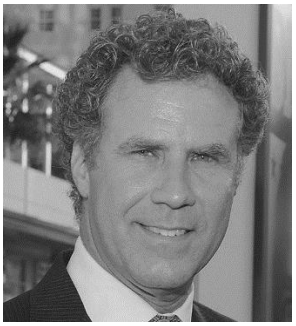


Figura 14. Imagen blanco y negro

Luego veremos cada píxel en nuestra imagen de a uno por vez. Para cada píxel, queremos ver los píxeles que lo rodean directamente. Ver Figura 15.

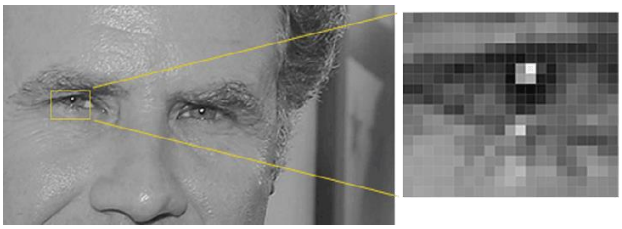


Figura 15. Pixel de una imagen

Nuestro objetivo es determinar cuán oscuro se compara el píxel actual con los píxeles que lo rodean directamente. Luego queremos dibujar una flecha que muestre en qué dirección la imagen se vuelve más oscura. Ver Figura 16.

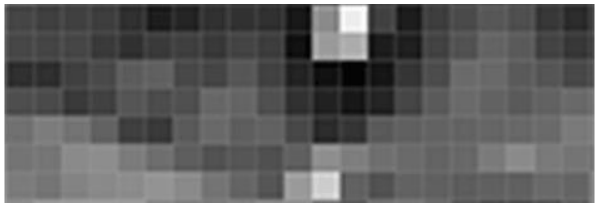


Figura 16. Dirección en que se vuelve más obscura la imagen

Al observar solo este píxel y los píxeles que lo tocan, la imagen se vuelve más oscura hacia la esquina superior derecha.

Si se repite ese proceso para cada píxel de la imagen, se termina con cada píxel reemplazado por una flecha. Estas flechas se llaman gradientes y muestran el flujo de claro a oscuro en toda la imagen. Ver Figura 17.

Figura 17. Gradientes de la imagen

Esto puede parecer algo aleatorio, pero hay una buena razón para reemplazar los píxeles con gradientes. Si analizamos los píxeles directamente, las imágenes realmente oscuras y las realmente claras de la misma persona tendrán valores de píxeles totalmente diferentes. Pero al considerar solo la dirección en que cambia el brillo, tanto las imágenes realmente oscuras como las realmente brillantes terminarán con la misma representación exacta. ¡Eso hace que el problema sea mucho más fácil de resolver!

Pero guardar el gradiente para cada píxel nos da demasiados detalles. Sería mejor si pudiéramos ver el flujo básico de luminosidad/oscuridad en un nivel más alto para poder ver el patrón básico de la imagen.

Para hacer esto, dividiremos la imagen en pequeños cuadrados de 16×16 píxeles cada uno. En cada cuadro, vamos a contar cuántos puntos de gradientes en cada dirección principal (cuántos apuntan hacia arriba, cuantos apuntan hacia la derecha, etc.). Luego reemplazaremos ese cuadrado en la imagen con las direcciones de flecha más fuertes.

El resultado final es que convertimos la imagen original en una representación muy simple que capta la estructura básica de una cara de una manera simple. Ver Figura 18.

Figura 18. Estructura básica de una cara

La imagen original se convierte en una representación HOG que captura las principales características de la imagen independientemente de los brillos de la imagen.

Para encontrar rostros en esta imagen de HOG, todo lo que tenemos que hacer es encontrar la parte de nuestra imagen que se parece más a un patrón de HOG conocido que se extrajo de un gran conjunto de otras caras de entrenamiento. Ver Figura 19.

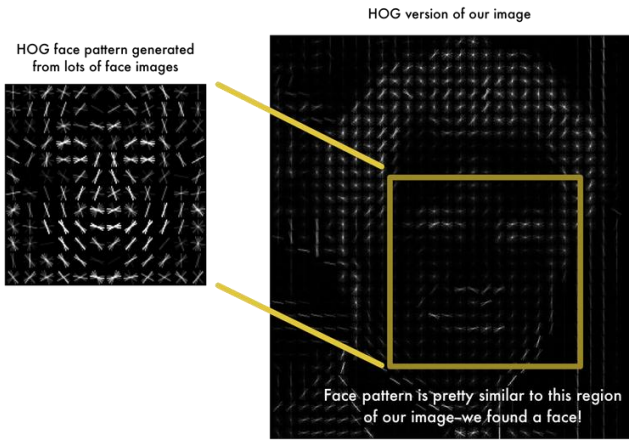


Figura 19. Patrón HOG

Usando esta técnica, ahora podemos encontrar caras fácilmente en cualquier imagen. Ver Figura 20.

Figura 20. Encontrando caras en una imagen

Extracción de características

Podemos especificar el número de orientación, pixels_per_celly cells_per_block para calcular las características HOG de un solo canal de una imagen. El número de orientaciones el número de contenedores de orientación que los gradientes de los píxeles de cada celda se dividirán en el histograma. El pixels_per_cellses el número de píxeles de cada fila y columna por celda sobre cada gradiente que se calcula el histograma. El cells_per_block especifica el área local sobre el cual se normalizaron los recuentos de histograma en una célula dada. Se dice que tener este parámetro generalmente conduce a un conjunto de características más robusto. También podemos usar el esquema de normalización llamado transform_sqrt que se dice que ayuda a reducir los efectos de las sombras y las variaciones de iluminación.

El reconocimiento de objetos en escenas desordenadas del mundo real requiere características de imágenes locales que no se ven afectadas por el desorden cercano u oclusión parcial Las características deben ser al menos parcialmente invariables para la iluminación, las transformaciones proyectivas en 3D y las variaciones comunes de objetos. Por otro lado, las características deben también sea lo suficientemente distintivo para identificar objetos específicos entre muchas alternativas La dificultad del problema de reconocimiento de objetos se debe en gran parte a la falta de éxito en encontrar tales características de imagen. Sin embargo, investigaciones recientes sobre el uso de características locales densas (por ejemplo, Schmid y Mohr [19]) ha demostrado que a menudo se puede lograr un reconocimiento eficiente mediante el uso de descriptores de imagen locales muestreados en un gran número de ubicaciones repetibles.

El reconocimiento de objetos se usa ampliamente en la industria de la visión artificial con fines de inspección, registro y manipulación. Sin embargo, los sistemas comerciales actuales para objetos reconocimiento depende casi exclusivamente de correlación basada comparación de plantillas. Si bien es muy efectivo para ciertos entornos de ingeniería, donde los objetos posan e iluminan están estrechamente controlados, la coincidencia de plantillas se vuelve computacionalmente inviable cuando la rotación de objetos, la escala, la iluminación, y la pose 3D pueden variar, y aún más cuando lidiar con visibilidad parcial y grandes bases de datos modelo.

Una alternativa a la búsqueda de todas las ubicaciones de imágenes para coincide es extraer características de la imagen que están en al menos parcialmente invariante para el proceso de formación de imágenes y coincidiendo solo con esas características. Muchas características candidatas se han propuesto y explorado tipos, incluidos segmentos de línea, agrupaciones de bordes y regiones, entre muchas otras propuestas. Si bien estas características tienen funcionó bien para ciertas clases de objetos, a menudo no se detectan con suficiente frecuencia o con suficiente estabilidad para formar Una base para el reconocimiento confiable

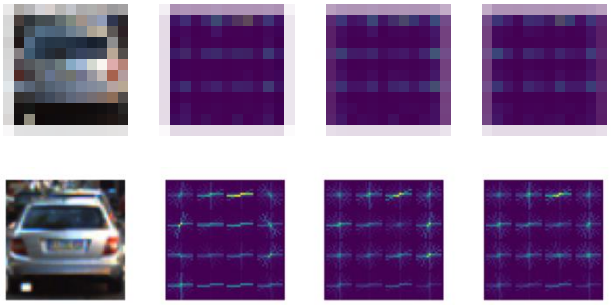


Figura 21. Reduciendo efectos de las sombras

CONCLUSIONES

Este trabajo de investigación se realizo con la finalidad de detectar objetos en una imagen sabiendo un poco mas a fondo las características de la imagen, el pixel y así poder llevar a cabo la realización de la detección que puede encontrarse en una imagen y ver su resultado para su visualización en el cual se basa en la visión artificial.

RECONOCIMIENTOS

A la facultad de Ingeniería de la Universidad Autónoma de Guerrero por dar las facilidades para poder obtener el titulo de

ingeniero en computación en el seminario de titulación en el cual se desarrollo este trabajo de investigación.

REFERENCIAS

[1] N. Dalal, Finding people in images and videos. PhD thesis, Institut National Polytechnique de Grenoble-INPG, 2006.

[2] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, vol. 1, pp. 886–893, IEEE, 2005.

[3] H. Cheng, N. Zheng, and J. Qin, "Pedestrian detection using sparse gabor filter and support vector machine," in Intelligent Vehicles Symposium, 2005. Proceedings. IEEE, pp. 583–587, IEEE, 2005.

[4] T. Joachims, Support Vector Machine, University of Dortmund, Informatik, AI-Unit Collaborative Research Center on 'Complexity Reduction in Multivariate Data', "http://svmlight.joachims.org/," 14.08.2008.

[5] L. G. Roberts, MACHINE PERCEPTION OF THREE-DIMENSIONAL soups. PhD thesis, Massachusetts Institute of Technology, 1963.

[6] K. K. Pingle, J. A. Singer, and W. M. Wichman, "Computer control of a mechanical arm through visual input.," in IFIP Congress (2), pp. 1563–1569, 1968.

[7] R. M. Haralick, K. Shanmugam, and I. H. Dinstein, "Textural features for image classification," Systems, Man and Cybernetics, IEEE Transactions on, no. 6, pp. 610–621, 1973.

[8] A. P. Witkin, "Recovering surface shape and orientation from texture," Artificial intelligence, vol. 17, no. 1, pp. 17–45, 1981.

[9] J. E. Mayhew and J. P. Frisby, "Psychophysical and computational studies towards a theory of human stereopsis," Artificial Intelligence, vol. 17, no. 1, pp. 349–385, 1981.

[10] B. K. Horn and B. G. Schunck, "Determining optical flow," in 1981 Technical Sym-posium East, pp. 319–331, International Society for Optics and Photonics, 1981.

[11] L. Kitchen and A. Rosenfeld, "Gray-level corner detection," Pattern recognition letters, vol. 1, no. 2, pp. 95–102, 1982.

[12] D. Marr and A. Vision, "A computational investigation into the human representation and processing of visual information," WH San Francisco: Freeman and Company, 1982.

[13] P. Viola and M. J. Jones, "Robust real-time face detection," International journal of computer vision, vol. 57, no. 2, pp. 137–154, 2004.

[14] D. G. Lowe, "Object recognition from local scale-invariant features," in Computer vision, 1999. The proceedings of the seventh IEEE international conference on, vol. 2, pp. 1150–1157, Ieee, 1999.

[15] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in Computer Vision–ECCV 2006, pp. 404–417, Springer, 2006.

[16] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 27, no. 10, pp. 1615–1630, 2005

[17] J. Canny, "A computational approach to edge detection," Pattern Analysis and Machine Intelligence, IEEE Transactions on, no. 6, pp. 679–698, 1986.

[18] W. T. Freeman and M. Roth, "Orientation histograms for hand gesture recognition," In International Workshop on Automatic Face and Gesture Recognition, vol. 12, pp. 296–301, 1995.

[19] J. Malik, S. Belongie, T. Leung, and J. Shi, "Contour and texture analysis for image segmentation," International journal of computer vision, vol. 43, no. 1, pp. 7–27, 2001.

[20] G. Wang, "A survey on training algorithms for support vector machine classifiers," in Networked Computing and Advanced Information Management, 2008. NCM'08. Fourth International Conference on, vol. 1, pp. 123–128, IEEE, 2008.

[21] V. Vapnik and S. Kotz, Estimation of dependences based on empirical data. Springer, 2006.

[22] J. Platt et al., "Sequential minimal optimization: A fast algorithm for training support vector machines," 1998.

[23] T. Joachims, "Making large scale svm learning practical," 1999.

[24] E. Osuna, R. Freund, and F. Girosi, "An improved training algorithm for support vector machines," in Neural Networks for Signal Processing [1997] VII. Proceedings of the 1997 IEEE Workshop, pp. 276–285, IEEE, 1997.