

**Detección visual automática de fuego por medio del Histograma de Gradientes  
Orientados y la Máquina de Vectores de Soporte**  
*Visual automatic fire detection through the Histogram of Oriented Gradients and Support  
Vector Machine*

**\* Cota, A., Cusi, E., Escobar, C. R., Orellana, F. K. y Rodríguez, P. R.**

**Carrera Ingeniería Informática e Ingeniería de Sistemas  
Facultad de Ciencias y Tecnología  
Universidad Mayor de San Simón  
Cochabamba - Bolivia**

Autor de correspondencia: \* contacto@scesi.org

**Resumen**

La detección visual de objetos en imágenes digitales, es uno de los temas desafiantes en la Ciencia de la Computación, más aún si dichos objetos no tienen una forma definida, como es el caso del fuego. Para resolver el desafío, es necesario recurrir a técnicas de la Visión por Computador y Aprendizaje Automático, de tal manera que los resultados se aproximen en lo posible, a los del sistema visual humano. Dada una imagen digital, con el objetivo de detectar las zonas donde existe fuego, el computador debería localizar exactamente las mismas zonas que un humano lo haría. Para un humano, la tarea de detección de objetos no implica mucha dificultad, en cambio para el computador es una tarea realmente difícil, puesto que, el fuego y los entornos donde podría manifestarse son infinitamente variables.

La importancia de la automatización de la detección visual de fuego, radica en la aplicación práctica que se puede dar, por ejemplo, el producto obtenido implantado en el computador de un drone, permitiría abarcar mayor territorio para la vigilancia de manifestaciones de fuego en zonas forestales.

En el presente artículo, se describe la implementación y evaluación de técnicas de la Visión del Computador y Aprendizaje Automático para la automatización de la detección visual de fuego en áreas forestales.

**Palabras clave:** detección de fuego; visión del computador; aprendizaje automático; HOG, SVM.

**Abstract**

*The visual detection of objects in digital images, is one of the challenging issues in Computer Science, especially if those objects do not have a definite shape, as in the case of fire. To meet the challenge, it is necessary to use techniques Computer Vision and Machine Learning, so that the results are approximate as much as possible, to the human visual system. Given a digital image, with the aim of identifying areas where fire there, the computer should locate exactly the same areas as a human would. For a human, the task of object detection does not involve much difficulty for the computer instead is a really difficult task, since, fire and environments which could manifest itself are infinitely variable.*

*The importance of automation in the visual detection of fire, lies in the practical application that can be given, for example, the product obtained implanted in the computer of a drone, would cover more territory for monitoring demonstrations of fire in forest areas.*

*In this article, implementation and evaluation of techniques Computer Vision and Automatic for the automation of visual detection of fire in forest areas described learning.*

**Keywords:** *fire detection; computer vision; machine learning; HOG, SVM.*

## Introducción

El fuego cuando se escapa de control, se convierte en un incendio, el incendio puede manifestarse en dos contextos: forestal o urbano (estructural). Un incendio forestal, es aquel donde el fuego y los fenómenos relacionados (humo, alta temperatura) se extienden en terrenos de vegetación. El cual, podría propagarse rápidamente, dependiendo del grado de inflamación de las especies forestales y de los factores climatológicos (vientos, sequía, altas temperaturas, etc.), lo que provoca un impacto negativo para el medio ambiente y los seres vivos que habitan a su alrededor.

Según el diario de noticias de circulación nacional Los Tiempos, Cochabamba perdió 5.400 hectáreas de áreas forestales por incendios en el año 2015, por 105 incendios registrados en 19 municipios, el 99 por ciento de los incendios fueron provocados por personas inexpertas en el uso de fuego para la preparación de tierras de cultivo (chaqueos) y la quema de basura. Ante esta alarmante situación, se manifiesta la importancia de contribuir con el desarrollo de tecnología informática, que coadyuve en zonas propensas a los incendios. Motivos por los cuales la principal interrogante que se pretende responder en la presente investigación es la siguiente:

¿Es posible automatizar la detección visual del fuego en imágenes tomadas desde un drone en vuelo en áreas forestales, por medio de la programación de técnicas de la Visión del Computador y del Aprendizaje Automático?

La programación de las técnicas para la detección de fuego, incorporadas en el computador de un aparato volador no tripulado (drone), permitirán vigilar y alertar oportunamente a los bomberos y a

otras entidades encargadas de conservar y mantener las áreas naturales de Cochabamba, es ahí donde radica la importancia práctica del presente trabajo. Por otro lado, el presente trabajo de investigación contribuye al área de la Ciencia de la Computación, en el sentido que las propuestas implementadas y evaluadas para la detección visual de fuego, podrían ser utilizadas para la detección de otros objetos amorfos en imágenes digitales, lo cual permitirá expandir nuevas áreas de investigación y de aplicación en nuestro medio.

## Métodos

El método empleado para la detección de fuego, comprende una serie de tareas que se pueden agrupar en tres partes, como se muestra en la Figura 1

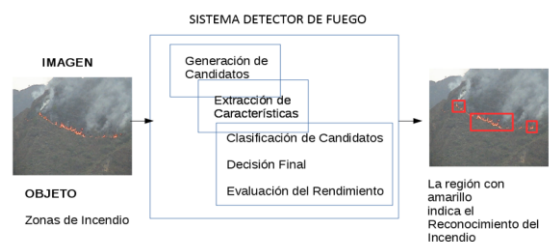


Figura 1: Esquema del sistema de la detección de fuego (Elaboración propia, 2016).

En las siguientes secciones, se explican los componentes del sistema detector de fuego.

### Generación de candidatos

En este paso se define las regiones de interés en la imagen, es decir, se generan las zonas que son candidatas a contener una instancia del objeto que se está buscando. Para ello se realizaron diferentes técnicas de visión por computadora, con la finalidad de determinar cuál de las ventanas (se entiende por ventana a la porción de imagen analizada) podrían ser potencialmente instancias del objeto que se

está buscando y cuáles se pueden descartar. De las distintas técnicas usadas, se tomaron en cuenta aquellas con las que se obtuvieron mejores resultados: segmentación por brillo, segmentación por color y difuminación.

**a) Segmentación por brillo:** la imagen obtenida en RGB<sup>11</sup> se pasó al modelo de color HSV<sup>12</sup>, con el fin de dividir la imagen en tres capas: H, S, V, para luego umbralizar la capa V obteniendo así una imagen binaria, con la que se consiguen todos aquellos contornos que pasan dicho umbral y por tanto, pasan a ser las regiones de interés.

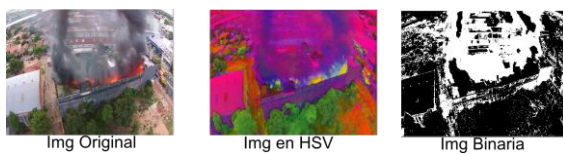


Figura 2: Segmentación por brillo

**b) Segmentación por color:** el algoritmo de rango de colores se aplica a la imagen, con el fin de obtener las regiones de interés que pasan por dicho rango y las que no pasan. Este algoritmo consiste en:

1. Pasar la imagen a HSV
2. Identificar los valores que corresponden a un máximo y un mínimo para delimitar las zonas de interés de la imagen HSV.
3. Obtener la máscara<sup>13</sup> resultante después de aplicar el rango de colores.

<sup>11</sup> RGB, del inglés Red, Green, Blue, traducido al español: Rojo, Verde, Azul.

<sup>12</sup> Modelo de color HSV. Del inglés, *Hue, Saturation, Value*, traducido al español: Matiz, Saturación, Valor.

<sup>13</sup> Conjunto de datos que aplicados a una imagen permite extraer ciertos datos de esta

4. Haciendo uso de la máscara encontrar los contornos de las regiones que pertenezcan al rango de colores.

**c) Difuminación:** el filtro Blur se aplica a la imagen, con el objetivo de desenfocar y reducir el ruido de la misma. Se debe tomar en cuenta el tamaño del kernel<sup>14</sup> ya que este es directamente proporcional a la difuminación en la imagen



Figura 3: Difuminación de una imagen usando un kernel de (15, 15)

### Extracción de características

La Extracción de características, es también conocida como el “Cálculo de descriptores”, en esta etapa se obtienen las características de las ventanas candidatas (o partes de la imagen), obtenidas en el paso anterior.

El descriptor que se ha utilizado es el Histograma de Gradientes Orientados, más conocido como HOG<sup>15</sup>, ya que se basa en la información del gradiente<sup>16</sup> de la imagen y reduce el número de dimensiones (número de variables), conservando la mayor cantidad de información.

<sup>14</sup> Matriz cuadrada de tamaño  $n \times n$  ( $n$  impar) utilizado en el proceso de filtrado

<sup>15</sup> HOG, del inglés Histogram of Oriented Gradient.

<sup>16</sup> Cambio direccional en la intensidad de la imagen

HOG es un descriptor de la imagen que utiliza el gradiente en cada uno de los pixeles como información básica.

El cálculo del gradiente, se basa en el cambio de intensidad de los pixeles adyacentes en dirección horizontal y vertical de la imagen

$$dy = I(x, y+1) - I(x, y-1)$$

$$dx = I(x+1, y) - I(x-1, y)$$

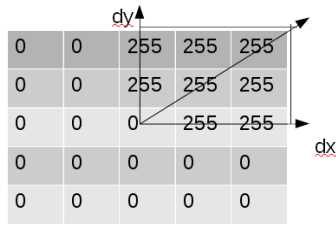


Figura 3: Cálculo de la diferencia de intensidad entre píxeles

Utilizando estas diferencias se calcula la dirección ( $\theta$ ) y magnitud ( $g$ ) del vector

$$\Theta(x, y) = \arctan\left(\frac{dy}{dx}\right)$$

$$g(x, y) = \sqrt{dx^2 + dy^2}$$

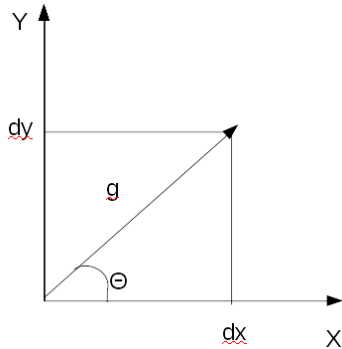


Figura 4: Gráfica de la relación entre la magnitud y dirección

Para imágenes a color se calcula el gradiente para cada uno de los tres canales: R, G, B y se selecciona el gradiente del canal con magnitud mayor.

Para realizar el cálculo se divide la región de interés en  $n \times n$  celdas de igual tamaño, para cada una de estas se calcula el histograma de orientaciones. Cada histograma permite obtener la información de las orientaciones dominantes en la imagen, que corresponderá a los valores altos, también obtiene la información de cómo están distribuidos los gradientes. Generando así el descriptor final, que es el resultado de la concatenación de histogramas de todas las celdas.

Combina la información del gradiente en histogramas locales que se calculan en celdas<sup>17</sup> de tamaño pequeño que se distribuyen por toda la imagen. Cada uno de los histogramas de orientación permite obtener la información de las orientaciones dominantes en la imagen, que corresponderá a los valores altos en las imágenes, también obtiene la información de cómo están distribuidos los gradientes en la imagen.

Los histogramas que se calculan localmente en diferentes posiciones de la imagen se agrupan en bloques<sup>18</sup> de tamaño mayor. Los bloques sirven para normalizar la representación final y para hacerla invariante a cambios de iluminación y distorsiones en la imagen.

El tamaño del descriptor final depende de algunos parámetros:

- $t$  = tamaño de la celda
- $signo$  = signo del gradiente, 180 sin signo y 360 (con signo)

<sup>17</sup> Agrupación de píxeles

<sup>18</sup> Agrupación de varias celdas vecinas

- $n^{\circ} \text{ intervalos} = \frac{\text{número de intervalos del histograma de orientaciones}}{\text{de orientaciones}}$
  - $n^{\circ} \text{ celdas/bloque} = \frac{\text{número de celdas por bloque}}{\text{de orientaciones}}$
  - $n^{\circ} \text{ celda} = \text{tamaño de la celda}$
  - $n = \text{tamaño del descriptor}$
- $\text{número bloques} = n^{\circ} \text{ celdas} - n^{\circ} \text{ celdas/bloque} + 1$   
 → Calcular en el eje x y y

$$n = \text{número bloques}_x \times \text{número bloques}_y \times n^{\circ} \text{ celdas/bloque} \times n^{\circ} \text{ intervalos}$$

### Clasificación de candidatos, decisión final y evaluación de rendimiento

Para clasificar los candidatos se utilizó la Máquina de Vectores de Soporte, más conocido como SVM<sup>19</sup>, debido a que, es eficiente tanto para el entrenamiento como para la clasificación, además es robusto en la generalización.

SVM es un clasificador lineal basado en el margen máximo a partir de los vectores de soporte. Al decir clasificador lineal se refiere a que existe una única frontera de separación entre dos clases representada por una línea en  $\mathbb{R}^2$  y por un hiperplano en el caso general de  $\mathbb{R}^n$ , para obtener el hiperplano considera algunas muestras del conjunto de entrenamiento (los vectores de soporte), estos vectores de soporte son elegidos de manera que la distancia o margen ( $d_+$ ,  $d_-$ ), entre los hiperplanos a la muestra positiva y negativa sea máxima.

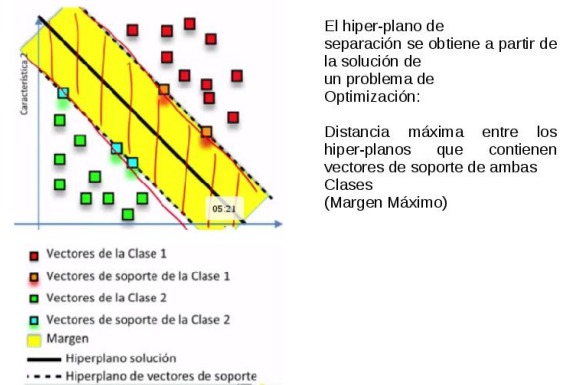


Figura 5: Esquema del cálculo de SVM

Para que el SVM pueda ser eficiente en conjuntos no linealmente separables:

- Se relaja la condición del margen (margen suave): Esto implica añadir una tolerancia a errores (vectores violando la condición de margen). Esta tolerancia está controlada por las slack variables<sup>20</sup>.
- Se transforma el espacio de características en otro linealmente separable (kernel trick): Los conjuntos no linealmente separables pueden transformarse en linealmente separables en un espacio de dimensión superior. en esta transformación no es necesario definir la función de mapeo entre el espacio original y el espacio superior, sólo será necesario definir el producto escalar (kernel).

Para la detección se probó kernels lineales y no lineales, obteniendo un mejor resultado con el kernel no lineal: Kernel Gaussiano de Base Radial (RBF).

Para calcular el kernel, se define la siguiente fórmula:

$$K(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2}, \gamma > 0$$

<sup>19</sup> SVM, el inglés Support Vector Machine

<sup>20</sup> Variables que permiten relajar la condición de margen máximo

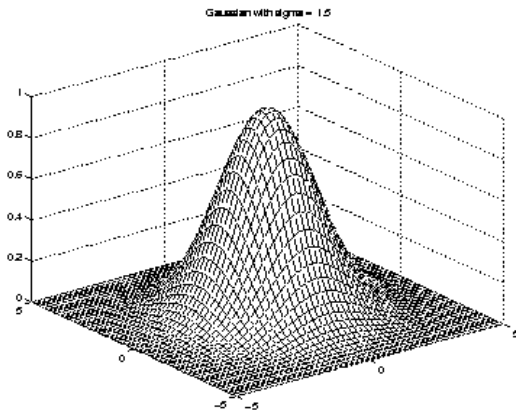


Figura 6: Kernel de Función de Base Radial (Chin, 1998)

Este kernel es de uso general y da buen resultado en la mayoría de los casos. El exponencial define una distancia entre las 2 muestras a partir de una función gaussiana.

Para validar los parámetros del kernel, se probó los valores recomendados por Hsu *et. al.* (2016)., para los parámetros:  $C$  (factor de regularización) y  $\gamma$  (gamma) con la validación “hold-out”<sup>21</sup>.

Los valores recomendados para estos parámetros son secuencias exponenciales.

$$C = 2^{-5}, 2^{-3}, \dots, 2^{15}$$

$$\gamma = 2^{-15}, 2^{-13}, \dots, 2^3$$

Combinando los parámetros  $C$  y  $\gamma$  se realizó varios entrenamientos y para validar estos entrenamientos se calculó el porcentaje de error, para lo cual, se utilizó un 20% del conjunto de candidatos y reales negativos. Del 20%, 10% es parte del conjunto de entrenamiento y los restantes 10% corresponde al conjunto de prueba, es decir, aquellas imágenes que no formaron parte del entrenamiento.

<sup>21</sup> Método Hold-out (método de retención): técnica usada para entrenar las muestras y evaluar la capacidad de generalización de dichas muestras

Se eligió aquellos valores de los parámetros con los que se obtuvo el mínimo porcentaje de error en la detección aplicando diferentes pre-procesamientos que fueron descritos anteriormente.

## Resultados

La siguiente tabla muestra los resultados obtenidos luego de aplicar las diferentes técnicas de preprocesamiento a las imágenes, tanto en la etapa de entrenamiento como en la etapa de detección. Los porcentajes de error corresponden a aquellos entrenamientos que tiene una mayor exactitud en la detección.

| Propuesta | Pre-procesamiento aplicado en el entrenamiento | Pre-procesamiento en la detección     | % de error |
|-----------|------------------------------------------------|---------------------------------------|------------|
| 1         | Ninguno                                        | Segmentación por brillo               | 17.5 %     |
| 2         | Segmentación por color                         | Segmentación por color                | 10.38 %    |
| 3         | Segmentación por color y difuminación          | Segmentación por color y difuminación | 6.94 %     |

Tabla 1. Resultados de los diferentes entrenamientos



Figura 7: Resultado de la detección aplicando la propuesta 1 de la Tabla 1.



Figura 8: Resultado de la detección aplicando la propuesta 2 de la Tabla 1.



Figura 9: Resultado de la detección aplicando la propuesta 3 de la Tabla 1

Luego de realizar pruebas en videos, se obtuvo los siguientes resultados:

- Propuesta 1: Se observó que a pesar de detectar la presencia de fuego también tendía a confundirse con regiones que presentaban alto brillo tales como el sol, reflejos de luz, es decir, regiones con alto nivel de iluminación.
- Propuesta 2: se noto una mejora en la detección respecto a la anterior técnica en zonas con alta iluminación ya que esta técnica se enfoca en el color característico del fuego. Sin embargo similar a la primera

técnica esta se confunde con aquellas regiones que tienen un color parecido al del fuego. Otro inconveniente es que es muy restrictiva, ya que no considera aquellas zonas donde los colores del fuego varían respecto al rango utilizado.

- Propuesta 3: al integrar la difuminación con el rango de colores se pudo observar una mejora con respecto a las anteriores técnicas, y a diferencia de la segunda técnica recupera más regiones de la imagen que podrían tener presencia de fuego.

### Discusión

Los resultados obtenidos muestran que es posible automatizar la detección visual del fuego aplicando técnicas de visión por computador y aprendizaje automático. Sin embargo estas técnicas y algoritmos que se aplicaron presentan sus limitaciones, por ejemplo al entrenar SVM con una cantidad considerable de datos (del orden de los millones de vectores de soporte), el tiempo de entrenamiento crece exponencialmente en función del número de vectores de soporte. Además, su procedimiento de optimización es difícil de implementar y presenta una alta carga computacional.

Al analizar las características que presentan las imágenes de fuego se determina dos propiedades principales:

- Color
- Brillo

En base a este análisis se implementaron diferentes técnicas de Visión por Computador, que permiten realzar aquellas zonas con dichas propiedades y con los cuales se obtuvo mejores resultados.

- La primera propuesta es recomendable para imágenes que presenten poco nivel de iluminación.
- Para la segunda propuesta se puede ampliar el rango de colores, con el objetivo de hacer que la detección sea menos restrictiva. Esto podría disminuir la precisión en la detección, y al mismo tiempo, capturaría más zonas que presenten fuego (falsos positivos).
- En cuanto a la tercera propuesta, se podría usar diferentes tipos de difuminación y también probar diferentes tamaños de kernel.
- Para el entrenamiento en SVM se podría usar diferentes tipos de kernel con sus respectivos parámetros.

### Conclusiones

El producto del trabajo realizado, es uno de los componentes de un Sistema de alerta temprana de incendios forestales utilizando cámaras y drones como agentes de vigilancia y control del entorno, iniciado en octubre del 2015 en la FCyT-UMSS, bajo la dirección del Ing. Eduardo Di Santi PhD.

Dada la importancia práctica de la automatización de la detección de fuego, será de orden prioritario integrar las técnicas programadas con los otros componentes del Sistema y probarlos en el dron en vuelo.

### Agradecimientos

Se agradece la dirección y las contribuciones del Ing. Eduardo Di Santi PhD, en la presente investigación.

Al Programa MEMI - UMSS, por el apoyo al grupo de investigación del Sistema de alerta temprana.

Al apoyo del Ing. Alexey Rodríguez por la donación de un dron para la investigación.

### Referencias

- Hsu Ch., Chang Ch. & Chih-Jen L. (2016). A Practical Guide to Support Vector Classification. Disponible en <http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf>
- K. K. Chin, (1998). Support Vector Machines applied to Speech Pattern Classification. Disponible en [http://svr-www.eng.cam.ac.uk/~kkc21/thesis\\_main/thesis\\_main.html](http://svr-www.eng.cam.ac.uk/~kkc21/thesis_main/thesis_main.html)
- López A., Valveny E. & Vanrell M. (2015), Detección de Objetos. Disponible en <https://es.coursera.org/learn/deteccion-objetos>

**Presentado:** Cochabamba, 16 de Septiembre de 2016.

**Aceptado:** La Paz, 10 de Octubre de 2016