



Institución Universitaria

Metodología para la identificación automática de flama/humo por medio de análisis de patrones dinámicos de imágenes digitales en entornos abiertos

Jimena Piedrahita Gil

Instituto Tecnológico Metropolitano

Facultad de Ingenierías

Medellín, Colombia

2017

Metodología para la identificación automática de flama/humo por medio de análisis de patrones dinámicos de imágenes digitales en entornos abiertos

Jimena Piedrahita Gil

Tesis o trabajo de investigación presentado como requisito parcial para optar al título de:
Magister en Automatización y Control Industrial

Director (a):

Ph. D María Constanza Torres Madroñero

Línea de Investigación:

Visión artificial y Fotónica

Grupo de Investigación:

Automática, Electrónica y Ciencias Computacionales

Instituto Tecnológico Metropolitano

Facultad de Ingenierías

Medellín, Colombia

2017

A Dios Padre, autor de toda patente natural del universo como creador y quien me permitió estar en un lugar privilegiado de la vida, quien, además, me proporcionó los medios intelectuales, físicos y espirituales para llevar a cabo este paso en mi vida.

A mis padres por darme la vida y su apoyo incondicional en cada escala y proyecto de mi vida.

A mi abuela a quien debo todo lo que soy.

A mi esposo y familiares por brindarme su paciencia, colaboración, apoyo y comprensión.

Agradecimientos

Al Omnisciente por levantarme en las caídas y darme desde su espíritu, su aliento, es el carisma que me ha hecho lograr cosas grandes en mi vida como mujer, como ingeniera, como hija como esposa como amiga.

A mi esposo, a mi asesora de tesis, a la empresa Accequip y finalmente a todas aquellas personas que con hechos me empujaron con sus consejos, paciencia.

Resumen

A pesar de los avances en el campo de la visión artificial, los métodos computacionales para la detección de incendios desde imágenes en el rango espectral visible aun presentan tasas de falsas alarmas significativas. En este proyecto se propone una metodología para la detección automática de incendios en el rango espectral visible, a partir de la extracción de características para discriminar eventos de flama/humo y la selección de técnicas para el análisis y procesamiento de secuencias de imágenes. Se compararon varios modelos de color para establecer el espacio de representación más apropiado para la detección de flama/humo. Secuencias de imágenes se segmentaron con técnicas de detección de movimiento aplicando dos configuraciones: una topología en cascada y otra en paralelo, mezclando segmentación por movimiento y color. Para la segmentación por movimiento se usan dos funciones de Matlab, la función Riff que identifica las diferencias en valor absoluto de los frame y la función Bitxor que identifica la diferencia binaria entre los frame. Para la segmentación por color, se exploraron siete modelos de color RGB, YCbCr, CMY, CMYK, HSI, HSL, y HSV. Finalmente, con las imágenes segmentadas se generó una matriz de características a partir de las cuales se entrenó un clasificador y se realizó validación cruzada para determinar la mejor metodología para la detección de flama/humo en una secuencia de imágenes. Los resultados obtenidos demuestran que la detección de fuego en secuencias de imágenes logra una precisión del 93.1% al utilizar el modelo de color CMYK, usando la técnica de detección de movimiento Bitxor en la topología en paralelo.

Palabras clave:

Detección incendio por video VDF, Segmentación, topología cascada, topología paralelo y espacio color, función Riff función Bitxor.

Abstract

Despite the advances on the field of computer vision, the computational methods for video fire detection in the visible spectral range still have significant false alarm rates. This project proposes a methodology for automatic fire detection in the visible spectral range, through feature extraction to discriminate flame/smoke and the selection of techniques for analysis and processing of sequences of images. Several color models were compared to establish the most suitable representation space for flame/smoke detection. Image sequences were segmented with motion detection techniques by applying two configurations: cascade and parallel topologies, using both motion and color segmentation. For motion segmentation, this work used two Matlab functions, Diff function identifies the absolute difference value and Bitxor function identifies the binary difference among frames. For the color segmentation, this project explored seven color model RGB, YCbCr, CMY, CMYK, HIS, HSL and HSV. Finally, a matrix of features was generated from the segmented images, a classifier was trained and cross validation was performed to determine the best methodology for flame/smoke detection from a sequence of images. The obtained results showed that fire detection in image sequences achieves a precision of 93.1% when using the CMYK color model and Bitxor motion detection technique in the parallel topology.

Keywords:

Video Fire Detection VDF, Segmentation, waterfall topology, parallel topology and color space, Diff function Bitxor function.

Contenido

	Pág.
1. Introducción	1
1.1 Justificación.....	2
1.2 Descripción del problema	4
1.3 Objetivos	5
1.3.1 Objetivo general.....	5
1.3.2 Objetivos específicos.....	5
2. Estado del Arte	7
2.1 Procesamiento digital de imágenes.....	7
2.2 Detección de flama y humo en entornos abiertos	9
2.3.1 Extracción de características	11
2.3.2 Análisis de movimiento y parpadeo.....	14
2.3.3 Segmentación.....	15
2.3.4 Clasificación o detección	16
3. Marco Experimental	21
3.1 Base de datos	21
3.3.1 MIVIA.....	21
3.3.2 Viratdata	22
3.3.3 Videos de fuego de Youtube.....	23
3.2 Metodología propuesta.....	24
3.3 Comparacion de técnicas	25
3.3.1 Topología en Cascada.....	27
3.3.2 Topología en Paralelo.....	28
3.4 Extraccion de características.....	29
3.5 Clasificacion	30
4. Resultados.....	32
4.1 Comparacion de tecnicas	32
4.2 Extracción de características.....	51
4.3 Clasificacion.....	51
4.3.1 Método de validación cruzada	51
5. Conclusiones y recomendaciones.....	73
5.1 Conclusiones.....	73
5.2 5.2 Recomendaciones.....	73

Lista de figuras

Figura 1. Etapas del procesamiento digital de imágenes (Fuente propia).....	7
Figura 2. Metodología en video fire detection propuesta.	25
Figura 3. Topología en Cascada.	28
Figura 4. Topología en Paralelo.	29
Figura 5. Resultados de la topología de segmentación en cascada.	34
Figura 6. Resultados de la topología de segmentación en paralelo.	35
Figura 7. Resultados topología cascada vs topología en paralelo en espacio de color RGB	36
Figura 8. Topología cascada en espacio de color RGB	37
Figura 9. Topología paralelo en espacio de color RGB.....	37
Figura 10. Resultados topología cascada vs topología en paralelo en espacio de color YCBCR	38
Figura 11. Topología cascada en espacio de color YCBCR	39
Figura 12. Topología paralelo en espacio de color YCBCR	39
Figura 13. Resultados topología cascada vs topología en paralelo en espacio de color CMY	41
Figura 14. Topología cascada en espacio de color CMY	41
Figura 15. Topología en paralelo en espacio de color CMY	41
Figura 16. Resultados topología cascada vs topología en paralelo en espacio de color CMYK.....	42
Figura 17. Topología cascada en espacio de color CMYK	43
Figura 18. Topología paralelo en espacio de color CMYK	43
Figura 19. Resultados topología cascada vs topología en paralelo en espacio de color HSV.....	44
Figura 20. Topología cascada en espacio de color HSV	45
Figura 21. Topología paralelo en espacio de color HSV	45
Figura 22. Resultados topología cascada vs topología en paralelo en espacio de color HSL	46
Figura 23. Topología cascada en espacio de color HSL.....	47
Figura 24. Topología paralelo en espacio de color HSL	47
Figura 25. Resultados topología cascada vs topología en paralelo en espacio de color HSI	48
Figura 26. Topología cascada en espacio de color HSI.....	48
Figura 27. Topología paralelo en espacio de color HSI	49

Lista de tablas

	Pág.
Tabla 1. Resumen de trabajos previos para la caracterización de flama/humo en imágenes y video en el espectro visible.	11
Tabla 2. Resumen de métodos de clasificación empelados para la detección de flama/humo en imágenes y video en el espectro visible.	16
Tabla 3. Descripción videos Mivia	21
Tabla 4. Descripción videos Virat	22
Tabla 5. Videos de flama seleccionados de la plataforma libre youtube	23
Tabla 6. Reglas de espacio de color	26
Tabla 7. Porcentaje de detección de la topología de segmentación en cascada	33
Tabla 8. Resultados de la topología de segmentación en paralelo	34
Tabla 9. Resultados topología cascada vs topología en paralelo en espacio de color RGB	36
Tabla 10. Resultados topología cascada vs topología en paralelo en espacio de color YCBCR.	38
Tabla 11. Resultados topología cascada vs topología en paralelo en espacio de color CMY	40
Tabla 12. Resultados topología cascada vs topología en paralelo en espacio de color CMYK	42
Tabla 13. Resultados topología cascada vs topología en paralelo en espacio de color HSV	44
Tabla 14. Resultados topología cascada vs topología en paralelo en espacio de color HSL	45
Tabla 15. Resultados topología cascada vs topología en paralelo en espacio de color HSI	47
Tabla 16. Resultados topología cascada vs topología en paralelo en los 7 espacios de color	49
Tabla 17. Descripción de parámetros en tablas de resultados.	51
Tabla 18. Resultados en RGB para la base de datos de 38 videos	52
Tabla 19. Resultados en YCBCR para la base de datos de 38 videos	53
Tabla 20. Resultados en CMY para la base de datos de 38 videos	53
Tabla 21. Resultados en CMYK para la base de datos de 38 videos	54
Tabla 22. Resultados en HSV para la base de datos de 38 videos	54
Tabla 23. Resultados en HSL para la base de datos de 38 videos	55
Tabla 24. Resultados en HSI para la base de datos de 38 videos	55
Tabla 25. Resultados en RGB para la base de datos de 38 videos en la matriz de videos originales	56
Tabla 26. Resultados en RGB para la base de datos de 38 videos en la matriz de videos segmentados	57
Tabla 27. Resultados en YCBCR para la base de datos de 38 videos en la matriz de videos originales	57

Tabla 28. Resultados en YCBCR para la base de datos de 38 videos en la matriz de videos segmentados	58
Tabla 29. Resultados en CMY para la base de datos de 38 videos en la matriz de videos originales.....	59
Tabla 30. Resultados en CMY para la base de datos de 38 videos en la matriz de videos segmentados.....	59
Tabla 31. Resultados en CMYK para la base de datos de 38 videos en la matriz de videos originales.....	60
Tabla 32. Resultados en CMYK para la base de datos de 38 videos en la matriz de videos segmentados	60
Tabla 33. Resultados en HSV para la base de datos de 38 videos en la matriz de videos originales.....	61
Tabla 34. Resultados en HSV para la base de datos de 38 videos en la matriz de videos segmentados.....	61
Tabla 35. Resultados en HSL para la base de datos de 38 videos en la matriz de videos originales.....	62
Tabla 36. Resultados en HSL para la base de datos de 38 videos en la matriz de videos segmentados.....	63
Tabla 37. Resultados en HSI para la base de datos de 38 videos en la matriz de videos originales.....	63
Tabla 38. Resultados en HSI para la base de datos de 38 videos en la matriz de videos segmentados.....	64
Tabla 39. Resultados en RGB para la base de datos de videos exceptuando los videos de flama azul en la matriz de videos originales.....	64
Tabla 40. Resultados en RGB para la base de datos de videos exceptuando los videos de flama azul en la matriz de videos segmentados.	64
Tabla 41. Resultados en RGB para la base de datos de videos exceptuando los videos de flama azul en la matriz de videos originales y segmentados unidos.	65
Tabla 42. Resultados en YCBCR para la base de datos de videos exceptuando los videos de flama azul en la matriz de videos originales.....	65
Tabla 43. Resultados en YCBCR para la base de datos de videos exceptuando los videos de flama azul en la matriz de videos segmentados.	65
Tabla 44. Resultados en YCBCR para la base de datos de videos exceptuando los videos de flama azul en la matriz de videos originales y segmentados unidos.	66
Tabla 45. Resultados en CMY para la base de datos de videos exceptuando los videos de flama azul en la matriz de videos originales.....	66
Tabla 46. Resultados en CMY para la base de datos de videos exceptuando los videos de flama azul en la matriz de videos segmentados.	66
Tabla 47. Resultados en CMY para la base de datos de videos exceptuando los videos de flama azul en la matriz de videos originales y segmentados unidos.	66
Tabla 48. Resultados en CMYK para la base de datos de videos exceptuando los videos de flama azul en la matriz de videos originales.....	67
Tabla 49. Resultados en CMYK para la base de datos de videos exceptuando los videos de flama azul en la matriz de videos segmentados.	67

Tabla 50. Resultados en CMYK para la base de datos de videos exceptuando los videos de flama azul en la matriz de videos originales y segmentados unidos.....	67
Tabla 51. Resultados en HSV para la base de datos de videos exceptuando los videos de flama azul en la matriz de videos originales.....	68
Tabla 52. Resultados en HSV para la base de datos de videos exceptuando los videos de flama azul en la matriz de videos segmentados.....	68
Tabla 53. Resultados en HSV para la base de datos de videos exceptuando los videos de flama azul en la matriz de videos originales y segmentados unidos.....	68
Tabla 54. Resultados en HSI para la base de datos de videos exceptuando los videos de flama azul en la matriz de videos originales.....	69
Tabla 55. Resultados en HSI para la base de datos de videos exceptuando los videos de flama azul en la matriz de videos segmentados.....	69
Tabla 56. Resultados en HSI para la base de datos de videos exceptuando los videos de flama azul en la matriz de videos originales y segmentados unidos.....	69
Tabla 57. Resultados en HSL para la base de datos de videos exceptuando los videos de flama azul en la matriz de videos originales.....	70
Tabla 58. Resultados en HSL para la base de datos de videos exceptuando los videos de flama azul en la matriz de videos segmentados.....	70
Tabla 59. Resultados en HSL para la base de datos de videos exceptuando los videos de flama azul en la matriz de videos originales y segmentados unidos.....	70

Metodología para la identificación automática de flama/humo por medio de análisis de patrones dinámicos de imágenes digitales en entornos abiertos

1. Introducción

Los sistemas de detección de incendio para recintos y entornos abiertos se encuentran en desarrollo dado que aún presentan limitaciones en la efectividad para identificar flama/humo (Pevzner & Waterman, 2009). Las limitaciones que presentan estos sistemas provocan falsas alarmas, pánico entre las personas alertadas, poca confiabilidad en el buen funcionamiento del sistema y evacuaciones innecesarias, abriendo así las posibilidades para mejores alternativas (a. E. Çetin et al., 2013). Esta problemática genera pérdidas económicas debido a la activación errónea de agentes extintores, interrupciones de emergencia de maquinaria y procesos de producción, además, pánico entre las personas alertadas, evacuaciones innecesarias y poca confiabilidad en el buen funcionamiento del sistema (A. E. Çetin, 2016; Rafael C. Gonzalez, 1975).

Se han propuesto técnicas para la detección de fuego basadas en alguno de los fenómenos que le acompañan: gases, humos, temperaturas o radiación UV, visible o infrarroja (Bharathi, Gokilapriya, Elango, & Vidhya, 2016; a. E. Çetin et al., 2013; Delgado, Ruiz, Sánchez, & Vila, 2014; D. Zhang & Wang, 2016). Entre estas técnicas se encuentran soluciones basadas en inteligencia artificial que pueden ayudar a mejorar la precisión en detección de humo, que presenta variación de color, o mejorar los problemas de intermitencia en el espectro de luz de llamas caóticas.

Sin embargo, las técnicas computacionales, presentan aun limitaciones con altas tasas de falsas alarmas, que pueden ocasionar activaciones erróneas de actuadores de extinción o simplemente fallas en la detección oportuna, generando propagación de los incendios, pérdida de recursos naturales, económicos, hídricos y humanos (a. E. Çetin et al., 2013).

Dados los vacíos que presentan los sistemas en la detección oportuna de incendios, este proyecto de investigación explora la detección automática de eventos relacionados con

incendios en entornos abiertos, a través de análisis de patrones de movimiento y color en imagen/video en el rango espectral visible.

1.1 Justificación

Los sistemas de detección de incendios se utilizan en todo el mundo con el fin de proteger la vida y prevenir el daño de bienes físicos. En países de Europa como en Alemania, los incendios en viviendas particulares han dejado anualmente 600 muertos, 6.000 heridos y cerca de 800 millones de Euros en daños (Buemi et al., 2016; Chino, Avalhais, Jr, Traina, & Carlos, 2015).

Los incendios se dan tanto en recintos cerrados, en compartimientos, en espacios abiertos o en campos forestales. La detección de incendios en espacios abiertos es uno de los problemas más difíciles por ser fenómenos no lineales (H. Wang, Yang, Jiang, & Wang, 2008) y uno de los objetivos de las investigaciones actuales es la reducción de falsos positivos (Filizzola et al., 2016; Han & Bhanu, 2007; Kallur & Kallur, 2016; Kurup, 2014; Luo, Yan, Wu, & Zheng, 2015).

El humo es un factor de incendio y los detectores para esta característica presentan limitantes en la diferenciación de partículas de otros eventos, tales como el cigarrillo prendido, partículas generadas por la fricción de materiales o simplemente el polvo común de cualquier entorno (Derbel, 2004).

Un sistema contra incendio convencional, se integra de agentes extintores que son elementos finales que actúan para controlar el fuego, estos agentes se componen de CO₂ (Bharathi et al., 2016; X. Wang, Lo, & Zhang, 2013), agua (a. E. Çetin et al., 2013; B. Ko, Jung, & Nam, 2014; J. Wang & Wang, 2014), espuma (Derbel, 2004; B. Ko, Cheong, & Nam, 2010; Y. Wang, Yu, Tu, & Zhang, 2011), entre otros. La aplicación de cada agente depende de la especialidad de protección a los bienes y personas que se deben salvaguardar, algunos de estos, son:

- El CO₂ es utilizado para extinguir incendios en recintos donde hay equipos eléctricos, debe ser descargado en áreas cerradas según la norma NFPA (A. K. K. Wong & Fong, 2014) para garantizar la extinción del incendio, pero en estas áreas no debe haber personas dentro del recinto en el momento de una descarga ya que este gas puede afectar la salud y hasta la vida.
- El agua es utilizada para extinguir incendios en recintos como oficinas, bodegas o espacios libres, en algunos casos puede afectar materias primas, maquinaria en producción, productos terminados, costos de limpieza y recuperación.
- La espuma es constituida de soluciones acuosas utilizado como agente de prevención y control de incendios para riesgos de líquidos inflamables.

La activación de algunos agentes extintores pone en riesgos a las personas, por consiguiente, las falsas alarmas en los sistemas contra incendio también cobran víctimas. Por ejemplo, en Estados Unidos, 29 bomberos han perdido la vida en los últimos 10 años por eventos tales como derrame cerebral, ataque cardíaco, colisión de vehículos, entre otros, atribuidos a los sistemas de extinción (Basset Blesa, 2000a).

En 2003, el Sistema Nacional de Informes sobre Incidentes por Incendios, la Administración de Incendios de los EE.UU, incluyó datos sobre falsas alarmas, provocadas por mal funcionamiento (como la detección errónea de humo) y activaciones no intencionales. Estos eventos representaron tres cuartos de todas las respuestas a las mismas. Las activaciones no intencionales de detectores de humo provocaron una de cinco alarmas.

Es así como la detección automática de incendios se encuentra en desarrollo y aun es objeto de estudio reducir las falsas alarmas de los nuevos métodos computacionales para mejorar su efectividad. La caracterización de partículas de llama en espacios abiertos, orientado al análisis exploratorio de métodos eficientes con nuevas metodologías es muy prometedor (A. E. Çetin, 2016).

1.2 Descripción del problema

La detección de incendios comercialmente se da con tecnologías electrónicas convencionales a través de sensores de humo y/o llama, con métodos inteligentes de detección automática por visión artificial y por último, con métodos híbridos que combinan las dos metodologías anteriores.

El método con sensores es el más usado, pero es limitado, debido a que solo detectan zonas en situaciones puntuales. Adicionalmente, estos sensores requieren de una gran proximidad con el fuego y la mayoría de ellos no pueden proporcionar información sobre dimensión y otros datos como la evolución del incidente. Estos métodos presentan falsas alarmas por que confunden otras partículas con el humo y finalmente presentan retardos; por ejemplo, la distancia entre las partículas de carbono con el detector, (a. E. Çetin et al., 2013; Verstockt, 2011).

Detectar polvo en vez de humo es una gran limitante de los sensores, y por ende, produce falsas alarmas. Para contrarrestar este problema, se combinan dos o más sensores para la detección de humo y la detección de llama por CO² (S.-J. Chen, Hovde, Peterson, & Marshall, 2007). Sin embargo, los problemas persisten por la baja eficiencia en tiempo y confiabilidad en la detección.

La detección por imagen – video es una alternativa que se ha venido desarrollando desde los años 90 (S.E.Memane, 2015), presenta menos limitantes que las técnicas de detección por sensores. La combinación de métodos de análisis de imágenes digitales es un proceso que disminuye en gran cantidad las falsas alarmas. Como ventaja, se aprovecha que el humo se esparce rápidamente y que muchas veces en los incendios se observa el humo, pero no se observa ninguna llama. Sin embargo, en la noche no es posible observar humo, pero si es mucho más sencillo detectar llamas. Las cámaras por su emulación del ojo humano podrían detectar el humo o el fuego con mayor precisión que los sensores (a. E. Çetin et al., 2013).

En la literatura se encuentran cada vez la combinación de métodos de análisis de imágenes de video para la detección del humo y el fuego (Yadav, Gupta, Gaur, & Bhattacharya, 2012). Como ejemplo, el análisis del color por sí solo no podría detectar incendios porque todas las imágenes captadas que presenten color similar al fuego, podrían causar falsas alarmas por la variabilidad en el color, la densidad, la iluminación y el fondo (Oreyin, 2009). El análisis del movimiento por si solo también puede conducir a falsas alarmas, donde objetos de color parecido al fuego y un movimiento similar, pueden ser clasificados incorrectamente como fuego (Piccinini, Calderara, & Cucchiara, 2008). Es necesario delimitar también el análisis del movimiento en video para lograr sistemas más precisos. Otras limitantes al emplear estos métodos son también la iluminación, el clima, agentes externos, entre otros (A. E. Çetin, 2016).

Finalmente, los métodos inteligentes de detección automática por visión artificial son promisorios y en desarrollo, pero comparte algunas limitantes de los métodos convencionales como los falsos positivos, aunque de forma más reducida que con los sensores por lo que se está usando combinar los dos métodos en busca de mejorar la detección de incendios.

1.3 Objetivos

1.3.1 Objetivo general

Proponer una metodología para la detección automática de eventos relacionados con incendios en entornos abiertos por medio de análisis de patrones dinámicos en imagen/video en el rango espectral visible.

1.3.2 Objetivos específicos

- Realizar un análisis comparativo de diversas técnicas de análisis de imágenes digitales para la detección de eventos relacionados con incendios.

- Caracterizar las propiedades medibles en secuencias de imágenes digitales para la detección de eventos relacionados con incendios en el rango espectral visible.
- Establecer un espacio de representación que permita el análisis de patrones dinámicos en secuencias de imágenes digitales orientado a la selección de atributos discriminantes para la detección automática de eventos relacionados con incendios en entornos abiertos.

Metodología para la identificación automática de flama/humo por medio de análisis de patrones dinámicos de imágenes digitales en entornos abiertos

2.Estado del Arte

En este capítulo se presenta una revisión de los principales aspectos relacionados con el procesamiento de imágenes para la detección de flama y humo en entornos abiertos. Se presenta una revisión detallada de trabajos previos en el cual se emplean imágenes en el rango espectral visible para la detección de flama y humo, resaltando las principales técnicas empleadas para extracción de características y el proceso de detección.

2.1 Procesamiento digital de imágenes

El procesamiento digital de imágenes incluye técnicas y algoritmos que realizan un tratamiento digital de una o varias imágenes, con el fin de mejorar su apariencia o extraer información útil (características) empleando herramientas computacionales. Por lo general, las técnicas de procesamiento de imágenes tratan las imágenes como señales bidimensionales empleando métodos de procesamiento de señal. El procesamiento digital de imágenes se puede resumir en las siguientes etapas: Adquisición de video, pre-procesamiento, segmentación, extracción o selección de características y por último clasificación o detección (Gonzalez, 2002) (Ver Figura 1).

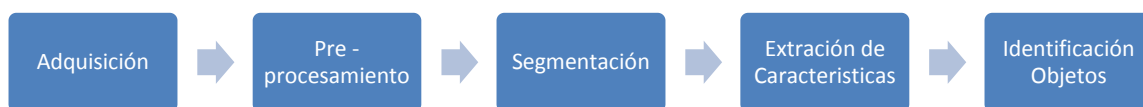


Figura 1. Etapas del procesamiento digital de imágenes (Fuente propia).

El preprocesamiento de imágenes según Gonzalez (2002) consiste en aplicar algoritmos que buscan mejorar la imagen original para resaltar determinadas características o para

eliminar otras que no son de interés para el problema. El preprocesamiento incluye técnicas de transformación de intensidad, ecualización del histograma, filtros de suavizado y filtros de realce.

La extracción o selección de características es opcional y dependen del problema (Gonzalez, 2002). Esta etapa tiene como objetivo la transformación de la imagen en un nuevo espacio de representación, obteniendo características estadísticas, de textura u otros con el fin de mejorar la representación de los objetivos y componentes de la imagen. La extracción o selección de características facilita la clasificación o detección de los objetos en la imagen. En la literatura se pueden encontrar diversas técnicas de extracción de características, tales como análisis de componentes principales (PCA), transformada Wavelet, descriptores de forma, medidas de textura, entre otros (a. E. Çetin et al., 2013).

La segmentación de imágenes busca dividir la imagen en regiones o segmentos manipulables. En este paso, se segmenta la imagen en regiones que tienen niveles de gris similares o en regiones donde existen cambios abruptos en los cambios del nivel de gris, detectando las fronteras o bordes de la imagen. En los sistemas de detección, una de las tareas con mayor dificultad es justamente la segmentación, pues requiere ser ejecutada en forma automática. Las técnicas de segmentación se pueden clasificar en métodos basados en detección de bordes, operaciones morfológicas, de umbral y crecimiento de regiones (Gonzalez, 2002).

La clasificación consiste en asignar el tipo de objeto al cual se encuentra asociado un pixel del espacio de representación. Por ejemplo, desde una imagen se puede obtener que pixeles corresponden a humo, fuego, edificios, vegetación, etc. Por su parte, la detección se puede ver como un problema de clasificación con dos clases, donde se determina la presencia o ausencia de una clase en particular. Los algoritmos de clasificación se dividen en dos categorías: clasificadores supervisados y clasificadores no supervisados. Los primeros realizan la clasificación usando información *a priori* para entrenar el sistema; entre los clasificadores supervisados más empleados se encuentran las máquinas de soporte vectorial (SVM por sus siglas en ingles Support Vector Machine), redes neuronales y máximo likelihood. Por su parte, los clasificadores no supervisados parten de medidas de similaridad para agrupar pixeles y encontrar las clases o modelo que define un conjunto

de variables; entre los clasificadores no supervisados se encuentran los algoritmos de clustering jerárquicos, particionales y basados en teoría de grafos (Sounthiraraj, Sahs, Greenwood, Lin, & Khan, 2014).

El desempeño de los clasificadores o detectores supervisados se realiza a través de una comparación cuantitativa entre el mapa de clasificación estimado y los datos de entrenamiento. Esta comparación se realiza a través de una matriz de confusión, la cual permite evaluar la precisión general de clasificación, la precisión de clasificación por clase y determinar que clases presentan mayor confusión (Gonzalez, 2002). Por su parte, los clasificadores no supervisados se evalúan a partir de medidas de similaridad inter e intra clases, lo cual permite determinar qué tan compactos y alejados se encuentran los grupos de clases encontrados por los algoritmos (Gonzalez, 2002).

El procesamiento digital de imágenes es usado generalmente en todas los campos de conocimiento como son las humanidades y ciencias sociales, la biología y biomedicina, los recursos naturales, las ciencias agrarias, la ciencia y tecnología de materiales, la ciencia y tecnología de alimentos, la ciencia y tecnologías químicas y la ciencia y tecnologías físicas (Gonzalez, 2002). Una de las líneas del procesamiento digital de imágenes es la detección de flama y humo en imágenes adquiridas en entornos abiertos (a. E. Çetin et al., 2013; Engineering, 2015).

2.2 Detección de flama y humo en entornos abiertos

La detección de incendios por imagen/video tiene sus orígenes en los años 90. Su desarrollo se ha dado gracias a la utilización y prestaciones de las cámaras de circuitos cerrados de Televisión CCTV, que registran imágenes en el rango espectral visible generalmente en alcances menores a 100 metros y en tiempo real (S.E.Memane, 2015). Mientras que la detección de incendios se realiza a través de técnicas que implementan sensores electrónicos, su uso se limita a espacios cerrados, presentando retardos y falsas alarmas. Sin embargo, la detección de incendio a partir de imágenes y video capturadas en el rango espectral visible por CCTV es un campo en desarrollo debido a que a través de estos datos se pueden monitorear grandes áreas en espacios cerrados y abiertos y,

con el avance en visión por computador esta detección se puede realizar en tiempo real (A. E. Çetin, 2016).

Según (a. E. Çetin et al., 2013) la detección automática de flama/humo a través de imágenes es un problema que se ha venido estudiando en el campo de la ingeniería y el área de la computación, con el objetivo de identificar de forma automática, precisa y oportuna incendios mejorando el desempeño de los métodos convencionales con sensores (S.E.Memane, 2015). Los incendios son provocados en dos tipos de entornos, en recintos abiertos y recintos cerrados (Lute, Sadaphal, Samudra, & Narkhede, n.d.). Se considera como entorno cerrado a un recinto como una habitación en el interior de un edificio. Y se considera entorno abierto a todo el ambiente exterior o a una gran edificación. Los incendios tienen características de llama y de humo diferentes dependiendo el tipo de entorno de propagación (Basset Blesa, 2000b; a. E. Çetin et al., 2013).

Cuando un incendio transcurre en un espacio abierto la llama crece sin ningún impedimento y se alimenta del aire en la medida que crece. La propagación de incendios en exteriores aumenta por la propagación del viento y la inclinación del terreno, lo que facilita el precalentamiento de los elementos combustibles por exposición (Basset Blesa, 2000b). Estas reacciones físicas determinan unas características de color, movimiento, entre otros que permiten identificar llama y humo de un incendio en espacio abierto a través de métodos computacionales.

El proceso de detección se realiza a través de técnicas de procesamiento digital de fotogramas que optimizan la imagen para la extracción o selección de características de la llama/humo. En el procesamiento de las imágenes, usualmente se realiza la segmentación a partir de estas características, y por último se emplean técnicas de clasificación que permitan la detección automática en tiempo real de incendios (A. E. Çetin, 2016).

2.3.1 Extracción de características

En la literatura se encuentran diversas técnicas para la extracción de características de fuego y de humo, las cuales aportan información importante para la detección automática de incendios. Los métodos para detectar incendios, se pueden distinguir en tres clases, dependiendo de las características analizadas: basada en el color, basada en el movimiento y basada en la textura (a. E. Çetin et al., 2013; Chino et al., 2015). La Tabla 2.1 resume diferentes métodos encontrados en la literatura para la caracterización y segmentación de imágenes y videos para la detección de flama/humo.

Tabla 1. Resumen de trabajos previos para la caracterización de flama/humo en imágenes y video en el espectro visible.

Características	Método	Trabajo
Espacio de color RGB y el canal R	Optimización de Enjambre de Partículas (PSO).	(Khatami, Mirghasemi, Khosravi, Lim, & Nahavandi, 2017)
Movimiento	Background sustraction Diff	(T. Ko, Soatto, & Estrin, 2008; Piccardi, 2004; Weihua, Lei, Junfeng, & Xinlong, 2011; Yadav et al., 2012)
Movimiento	Background sustraction Bitxor	(Shrivastava & Matlani, 2016)
Color RGB y brillo HIS Orientación.	Segmentación de umbrales y regiones	(Zheng, Yuan, & Lu, 2016)
Espacio de color RGB	técnica de sobel (Sobel Edge Detection)	(Paper, Poobalan, Universiti, & Liew, 2015)
Espacio de Color RGB, YCBCR, luminancia y crominancia de grises		(Bharathi et al., 2016)
Espacio de color RGB, en R	Umbrales	(Kallur & Kallur, 2016)
Espectros de iluminación	Cloudy-radiance-detection	(Filizzola et al., 2016)
Espacio de color RGB y tiempo y característica LDS.	Wavelet y análisis dinámico	(Mathi & Latha, 2016)
Color y textura	Segmentación de textura	(Chino et al., 2015)
Umbral	Segmentación de umbral de Otsu	(H. Zhang, Zhang, & Xiao, 2015)

Espacio de color RGB. Área, bordes, y fondo	Vectores de velocidad OMT	(Foggia, Saggese, & Vento, 2015)
Right-leaning line Smooth streamline Low-frequency Fixedsource Vertical-horizontal ratio		(Luo et al., 2015)
Tamaño, movimiento e información de color.		(S. G. Kong, Jin, Li, & Kim, 2016)
Espacio de color RGB y escala de grises		(Bandi & Patil, 2016)
Espacio de color RGB	Método de Blurring para eliminar ruido de color borroso	(Rajbhoj, Gholap, Dubale, & Gogawale, 2016)
Espacio de color YCbCr Parpadeo de llamas	Algoritmo de umbrales	(Yadav et al., 2012)
Espacio de color YCbCr	Segmentación	(Santana & Gomes, 2012)
Energía	Análisis de energía	(Piccinini et al., 2008; Zhao, Li, & Gu, 2015)
Espacio de color CMY	Segmentación	(En, Para, & Gratis, n.d.)
Espacio de color CMYK	Segmentación	(Szeliski, 2010)
Espacio de color CMYK	Segmentación	(En et al., n.d.)
Espacio de color HSV	Segmentación	(Lute et al., n.d.)
Espacio de color HSV, YCbCr y RGB	Segmentación	(Shi, Long, Zhan, & Lin, 2016)
Espacio de color HSL	Segmentación	(Lute et al., n.d.)
Espacio de color HSI	Segmentación	(Santana & Gomes, 2012)
Espacio de color HSI	Segmentación	(Jiayun, 2015)

El humo es difícil de detectar si se basa en características estáticas tales como color de humo, textura, etc., o en características dinámicas tales como frecuencia, forma o aleteo, etc. La detección de humo está siempre sujeta a algunas excepciones significativas, por ejemplo, el momento que el viento mueve una bolsa y esta tiene un movimiento en

balanceo, la conducción de vehículos especialmente en la noche, la niebla, e incluso personas en movimiento. (Luo et al., 2015).

Los métodos que utilizan el color se basan en la consideración de que una llama, bajo la suposición de que es generada por combustibles comunes como madera, plástico, papel u otro, se puede caracterizar de manera confiable por su color, de modo que la evaluación de los componentes de color (en RGB, YUV o cualquier otro espacio de color) es suficientemente robusta para identificar la presencia de llamas (A. E. Çetin, 2016). Algunos trabajos que emplean este enfoque son los trabajos realizados por (Chino et al., 2015).

Para la detección, se crea un modelo matemático basado en reglas, definiendo un sub-espacio en el espacio de color que representa todos los píxeles de la imagen. Hay varios modelos empíricos que utilizan diferentes espacios de color como RGB (Chino et al., 2015), YCbCr (Lute et al., n.d.), CIE (Kurup, 2014) y HSV (J. Chen, He, & Wang, 2010; Kurup, 2014; Lute et al., n.d.). En estos casos, la limitación es la falta de correspondencia de estos modelos para proyectar propiedades más allá del color.

El color no se puede utilizar por sí mismo para detectar fuego debido a la variabilidad del color, la densidad, la intensidad de iluminación o por los objetos de color con tonos y combinaciones rojo -amarillos, esto hace que se presentan falsas alarmas. Para este caso (Verstockt, 2011) utilizó la señal de vídeo que transporta la información de color de la imagen (crominancia) para contrastar esquemas de detección de humo por color. Para cualquier fuente de humo, el valor de la iluminación debe ser alto. Por otro lado, los valores de crominancia deben ser bajos (A. E. Çetin, 2016).

Los algoritmos de movimiento, de reconocimiento de fondo, de diferenciación de tiempo y de análisis de flujo óptico, se pueden utilizar como parte de un sistema de detección de fuego usando video.

En la extracción de características por movimiento la llama y el humo son objetos en movimiento. Para determinar si el movimiento es debido al humo u otro objeto ordinario es necesario un mayor análisis de las regiones en movimiento de video. Este enfoque de detección de llama por movimiento conduce a falsas alarmas por que la caída de hojas

secas o de objetos de color similar al de la llama, pueden ser clasificados erróneamente, la literatura deja abierto el campo para realizar más análisis de movimiento en video para lograr sistemas más precisos (Shrivastava & Matlani, 2016), (S.E.Memane, 2015; Yu, Mei, & Zhang, 2013).

Los algoritmos de reconocimiento de fondo han sido trabajados por (Celik, Demirel, Ozkaramanli, & Uyguroglu, 2007; J. Chen et al., 2010; Gubernov, Kolobov, Polezhaev, Sidhu, & Mercer, 2010; S. G. Kong et al., 2016; Lute et al., n.d.; Yuan, 2008; Zhao et al., 2015). En los algoritmos de fondo las cámaras se encuentran estáticas, un ejemplo es el trabajado referenciado por (A. E. Çetin, 2016) de Collin el cual utiliza dinámica de fondo basado en modelos de mezcla Gaussianos también utilizados por (J. Chen et al., 2010; Töreyn, Dedeoğlu, Güdükbay, & Çetin, 2006).

Los algoritmos de diferenciación de tiempos han sido trabajados por(S. G. Kong et al., 2016) y el análisis de flujo óptico, por (Kurup, 2014). La extracción de características de textura del fuego en imágenes fijas es un método alternativo al de movimiento, estos corresponden a mediciones numéricas que describen propiedades visuales. Estas propiedades son capaces de descubrir y representar las conexiones entre los píxeles de toda la imagen global, o de pequeñas regiones de la imagen local. Los descriptores de bajo nivel, como los basados en el color, la forma y la textura, se utilizan por (Chino et al., 2015).

2.3.2 Análisis de movimiento y parpadeo

Una de las características de las llamas en un incendio en propagación es el parpadeo. Estos rasgos se han trabajado con análisis de señales de energía (Luo et al., 2015; Piccinini et al., 2008; Zhao et al., 2015) en el dominio de Wavelet en (Frizzi & Kaabi, 2016; Gubbi, Marusic, & Palaniswami, 2009; Kandil, Salama, & Rashad, 2010; B. Ko et al., 2010; Luo et al., 2015; Töreyn et al., 2006; Vinicius, Borges, Izquierdo, & Member, 2010; Yuan, 2008, 2011; Zhao et al., 2015), estos métodos se centran en el comportamiento temporal de las llamas y el humo.

Los pixeles de los colores de la llama aparecen y desaparecen en los bordes turbulentos de estas. El libro de (A. E. Çetin, 2016) describe la investigación de dos trabajos donde se muestra experimentalmente que la frecuencia de parpadeo de una llama es de 10Hz, como resultado el uso de análisis de frecuencia para diferenciar las llamas de otros objetos se propone. Sin embargo, un fuego en etapa inicial exhibe un comportamiento caótico debido a que el proceso de combustión se compone de funciones no lineales debido a la intermitencia. Por lo tanto, las llamas turbulentas se pueden caracterizar como una actividad caótica con frecuencia de banda ancha. En efecto no es posible observar una sola frecuencia de parpadeo en el espectro de la luz. De igual manera pasa con el humo. Por lo tanto, las técnicas que analizan el parpadeo no son viables por si solas, pero pueden ser utilizadas en un sistema multifuncional de detección de fuego por video.

Los tiempos de Fourier, es una técnica utilizada para el análisis del parpadeo de llamas a frecuencias de 10Hz aunque no es necesario ya que el aumento de la Energía de Fourier en 550Hz es un factor de que hay llamas (B. Ugur Toreyin, Yigithan Dedeoglu, n.d.). De igual forma, además el comportamiento temporal del humo se puede tratar por análisis de energía en el dominio de Wavelet, el suavizado gradual de los bordes de la imagen y en esta, la variación del fondo de la imagen respecto a la imagen actual da como resultado presencia de humo (Gubbi et al., 2009).

2.3.3 Segmentación

La técnica de umbralización es utilizada por (Zheng et al., 2016) para segmentar las partes más brillantes de la imagen, con este resultado, la umbralización permite obtener regiones relevantes de una manera más rápida y eficiente. Con la segmentación se puede contar con todas las posibles características de color del fuego; sin embargo, este método presenta alta confusión con regiones que tienen niveles de brillo similares a los del fuego, como la luz solar, luz de focos, entre otros.

El patrón binario local (LBP) es un extractor de características de textura, el cual considera el entorno de un píxel para segmentar imágenes a través de un algoritmo de generación

de superpíxeles (Chino et al., 2015). Este método detecta bien los límites de la imagen de fuego y mejorar el rendimiento del algoritmo de segmentación.

El método de segmentación de umbral de Otsu designa la mayor varianza del método entre-clusters, es decir, selecciona un umbral denominado T para calcular la varianza de dos categorías divididas por T, para dejar que recorra todas las escalas de grises y seleccionar la escala de grises que se desea (H. Zhang et al., 2015).

2.3.4 Clasificación o detección

La detección automática de humo y de llama de un incendio se establece cuando un clasificador determina qué píxeles (pesos) del espacio de representación corresponden a una imagen de humo o a imágenes de llama. Por mencionar algunas técnicas para clasificar imágenes se encuentran las Máquinas de Soporte Vectorial SMV, las Redes Neuronales RN, y los Algoritmos de Clustering (Sounthiraraj et al., 2014).

Un método común para la detección de fuego o humo es SMV, a través de núcleos de base radial RBF. El problema del uso de esta máquina es que requiere de un gran número de secuencias (frames) de videos de fuego y de no fuego para reducir las falsas alarmas (Y. Zhang, Tian, & Liu, 2011).

Otro método usado para clasificar es el conocido como AdaBoost (Szeliski, 2010; Zhao et al., 2015), redes neuronales, clasificadores bayesianos, modelos de Markov y clasificación basada en reglas.

En la Tabla 2. Se puede observar el estado del arte en métodos de clasificación de llama o fuego

Tabla 2. Resumen de métodos de clasificación empelados para la detección de flama/humo en imágenes y video en el espectro visible.

<i>Método Clasificador</i>	<i>Trabajo</i>
K-medoids clustering y Otsu's thresholding algorithms.	(Khatami et al., 2017)
Maquina Soporte Vectorial SMV	(Zheng et al., 2016)
Máquina de soporte Vectorial SMV	(Mathi & Latha, 2016)
Segmentación umbrales y propagación del área	(Bharathi et al., 2016)
Robust Satellite Techniques for FIRES detection and monitoring - RST-FIRES – es una técnica de detección de cambios multitemporales	(Filizzola et al., 2016)
Sistema Multiexperto basado en Red Neuronal	(Foggia et al., 2015)
Técnica de condensación de imágenes	(Luo et al., 2015)
Regresión Logística y espacial	(S. G. Kong et al., 2016)
técnicas de detección de bordes de canny color y detección de movimiento	(Bandi & Patil, 2016)
Thresholding método de umbralizacion de OUTSO	(Rajbhoj et al., 2016)
Fuzzy Logic	(Buemi et al., 2016)
Transformada de Wavelet	(Yadav et al., 2012)

El método K-medoids clustering y Otsu's thresholding algorithms propuesto por (Khatami et al., 2017) detecta regiones de fuego con mayor precisión en comparación con otros métodos, incluso en imágenes de color complejas. Una razón importante que contribuye a esto es el aumento de contraste no lineal propuesto en su trabajo. La utilización de K-medoids en lugar de Fuzzy C-Means Clustering (FCM) es una ventaja. Porque K-medoids selecciona muestras de datos (medoides) como centros de agrupación, mientras que FCM considera un grupo por el valor medio (centroide) de sus muestras de datos asociadas. Además, la agrupación de K-medoides es eficaz en el manejo del ruido y los valores atípicos. En otras palabras, un medóide es apropiado para derivar una tendencia representativa de su muestra central, incluso en distribuciones sesgadas. De hecho, un medoid es una muestra de datos en el espacio de datos, mientras que un centroide puede estar fuera del espacio de datos. Por otro lado, la debilidad del método K-medoids clustering se puede evidenciar, cuando existe un alto grado de similitud de color entre los

diferentes componentes de una imagen. Esto hace que las regiones que no sean de fuego sean detectadas como fuego haciendo que se disminuya la precisión de detección.

La clasificación a través de núcleos de Función de Base Radial (RBF) es un enfoque trabajado por (Luo et al., 2015) y (Mathi & Latha, 2016) donde se usan vectores de características multidimensionales obtenidos de las imágenes para detectar llama o humo a través de SVM.

En el trabajo (Foggia et al., 2015) se han introducido dos novedades principales: en primer lugar, la información complementaria, basada respectivamente en el color, la variación de la forma y el análisis del movimiento con sistemas Multi expertos. La principal ventaja derivada de este enfoque reside en el hecho de que el rendimiento general del sistema aumenta respecto a otros enfoques que solo se basan en el contraste de las imágenes de fuego. En segundo lugar, ellos propusieron un nuevo descriptor basado en un conjunto de palabras para representar el movimiento. Aunque la restricción de límites tiene deficiencias en imagen borrosas.

La técnica de condensación de imágenes para detección de humo tratada por (Luo et al., 2015) presentó algunas deficiencias debido a que las trayectorias de humo distante y de movimiento lento son aproximadamente horizontales, pudiéndose considerar como objetos similares sin las características de la trayectoria del humo. En este caso, este método puede fallar en imágenes de humo con parámetros estáticos. Dejando abierto el estudio en cómo clasificar humo distante con movimiento lento o estática absoluta.

(S. G. Kong et al., 2016) trabajó detección de fuego mediante regresión logística y suavizado temporal en secuencias de imágenes detectadas por video de cámaras de seguridad. La probabilidad de incendio calculada mediante regresión logística muestra una alta sensibilidad a la detección de incendios considerandouna tasa media de detección de incendios de 98,5% con una velocidad de detección media de 16 tramas, lo que corresponde a aproximadamente 0,5 s.

Sin embargo, muchos objetos parecidos al fuego que tienen características similares a la llama de fuego pueden desencadenar una falsa alarma (Kurup, 2014; Paper et al., 2015) .

Esto es particularmente importante para la detección de incendios con cámaras de vigilancia de relativamente baja calidad. En este trabajo, el suavizado temporal se emplea para reducir significativamente las tasas de falsas alarmas, manteniendo al mismo tiempo las precisiones aceptables de detección de llama de fuego. Los resultados del experimento demuestran que el método propuesto funciona bien en varios ambientes interiores y exteriores con pocas falsas alarmas.

Los métodos de clasificación a través de Redes Neuronales, como los trabajados por (Mishra, Yadav, Ray, & Kalra, 2007), utilizaron un modelo basado en una sola neurona denominado como Integrate-and-Fire Neuron (IFN) logrando con este método detectar fuego, método que al ser comparado con una red perceptrón multicapa denominada como The Multilayer Perceptron (MLP), su aprendizaje fue mejor y los tiempos de entrenamiento y tiempos de prueba fueron significativamente menores. Por otro lado (Frizzi & Kaabi, 2016) trabajo igual con una red neuronal convolucional (CNN) para identificar el fuego en videos. La red neuronal convolucional funcionó muy bien en el área de clasificación de objetos. Esta red tiene la capacidad de realizar la extracción y clasificación de características dentro de la misma arquitectura. Probado en secuencias de vídeo reales. La matriz de confusión y las curvas ROC indican una muy buena precisión global para la etapa de detección.

Se anota que la CNN está actualmente limitada a manejar entradas 2D que nos lleva a procesar la entrada de vídeo sólo marco por cuadro. Por el contrario, la CNN 3D extrae características tanto espaciales como temporales mediante la realización de convoluciones 3D. Así, la información de movimiento de fuego y humo podría codificarse, lo que hace posible disminuir considerablemente el coste de tiempo. Además, para optimizar la detección y localización de humo y fuego en un video, se debe mejorar. Este trabajo sólo detecta el fuego rojo, para detectar otro color de fuego, tenemos que aumentar nuestro conjunto de entrenamiento con otros colores de fuego como el azul (Frizzi & Kaabi, 2016).

Por otro lado, el trabajo de (Buemi et al., 2016) se basó en un modelo de fuego difuso que permite reconocer áreas de color de fuego y en una prueba dinámica de movimiento para reducir falsos positivos. Los pasos principales son: La primera es la eliminación de fondo; La segunda es la Fuzzy Fire Search, que permite encontrar un conjunto de píxeles

potencialmente clasificables como fuego porque su color coincide con un modelo de color difuso que describe las características del fuego; El paso final es la Prueba Dinámica de Movimiento que permite reducir los falsos positivos analizando el movimiento de las áreas de color de fuego detectadas por la Fuzzy Fire Search. Este trabajo presenta pocos falsos negativos. Aunque el rendimiento de este algoritmo disminuye lentamente a medida que las condiciones de detección se degradan.

En los métodos de clasificación basados en reglas, Wavelet, Redes Neuronales, Bayesianos Modelos Markov, el preprocesamiento, las operaciones morfológicas, el subbloqueo son utilizados en sistema de VDF (Mathi & Latha, 2016; Rossi, Akhloufi, & Tison, 2011).

Hoy en video fire detection se están implementando técnicas con fines de reducir falsos positivos y falsos negativos en busca de obtener sistemas automáticos más eficientes y confiables, debido a que hay eventos que vía computacional se relacionan con eventos de incendio, como lo son las hojas en otoño al caer, las luces artificial en movimiento, entre otros (a. E. Çetin et al., 2013; J. Chen et al., 2010; H. Kong, 2010).

Se destaca finalmente en esta revisión que el color no se puede utilizar por sí mismo para detectar fuego debido a la variabilidad del color, la densidad, la intensidad de iluminación o por los objetos de color con tonos y combinaciones rojo -amarillos, hace que se presenten falsas alarmas. El método por textura del fuego en secuencias de imágenes es un método alternativo al de movimiento, estos corresponden a mediciones numéricas que describen propiedades visuales. Estas propiedades son capaces de descubrir y representar las conexiones entre los píxeles de toda la imagen global, o de pequeñas regiones de la imagen local. Los descriptores de bajo nivel, como los basados en el color, la forma y la textura, se utilizan por (Chino et al., 2015).

Con este trabajo se busca un nuevo método o la implementación de algún existente que procure mejoras en la respuesta a eventos de fuego buscando reducir falsos positivos y falsos negativos y aumentando el índice de precisión.

3.Marco Experimental

En este capítulo, se describen los métodos y técnicas trabajadas en el desarrollo de esta tesis, incluyendo la descripción de la base de datos y la metodología propuesta para la detección automática de eventos relacionados con incendios en entornos abiertos.

Como la interrelación del color no se puede utilizar por sí mismo se implementó para la detección de movimiento el método "Background Subtraction" el cual utiliza el movimiento y el color en un mismo procedimiento, sin embargo, a este método se le combinaron dos variantes utilizando el operador lógico XOR y otra con el valor absoluto de la diferencia de los frames. De esta manera en el método se contemplaron dos topologías de interconexión entre dos técnicas utilizadas para la detección de flamas (la detección de movimiento y el reconocimiento por color).

3.1 Base de datos

Los datos utilizados para el desarrollo de esta tesis se obtuvieron a partir de videos con imágenes de fuego y con ausencia de fuego, de eventos relacionados con incendios en entornos abiertos, extraídos de las plataformas de MIVIA, Viratdata y de Youtube, las tres de libre acceso.

3.3.1 MIVIA

MIVIA es una base de datos del laboratorio de investigación de la Universidad de Salerno – Italia activa en los campos de Reconocimiento de Formas y Visión por Computador. MiVIA Lab da contribuciones a aspectos teóricos de Reconocimiento de Patrones como paradigmas de clasificación sintáctica y estructural, a juego gráfico y el aprendizaje y la fiabilidad de clasificación a partir de bases de datos. La base de datos incluye 21 videos con precencia de fuego o humo tal como se describe en Tabla 3.

Tabla 3. Descripción videos Mivia

Video	Descripción
fire1	Si fuego Si humo
fire2	Si fuego No humo

fire3	Si fuego Si humo
fire4	Si fuego Si humo
fire5	Si fuego Si humo
fire6	Si fuego Si humo
fire7	Si fuego Si humo
fire8	Si fuego Si humo
fire9	Si fuego Si humo
fire10	Si fuego Si humo
fire11	Si fuego Si humo
fire12	Si fuego Si humo
fire13	Si fuego Si humo
fire14	Si fuego Si humo
fire15	No fuego Si humo
fire16	No fuego Si humo
fire17	No fuego Si humo
fire18	No fuego Si humo
fire19	No fuego Si humo
fire20	No fuego Si humo
fire21	No fuego Si humo

3.3.2 Viratdata

Viratdata es un conjunto de datos de referencia a gran escala para el reconocimiento de eventos en video de vigilancia realizado por el equipo de Viratdata integrado por Sangmin Oh, Anthony Hoogs, Amitha Perera, Naresh Cuntoor, Chia-Chih Chen, Jong Taek Lee, Saurajit Mukherjee, JK Aggarwal, Hyungtae Lee, Larry Davis , Xiaoyang Wang, Qiang Ji, Kishore Reddy, Mubarak Shah, Carl Vondrick, Hamed Pirsiavash, Deva Ramanan, Jenny Yuen, Antonio Torralba, Bi Song, Anesco Fong, Amit Roy-Chowdhury y Mita Desai, publicado en Proceedings of IEEE Computer Vision and Recognition Conference (CVPR). Realizado en el año 2011 (Viratdata solicitan referenciar así como aquí si es usada su base de datos). De esta base de datos se usaron 6 videos con la presencia de fuego o humo tal como se describe en la Tabla 4.

Tabla 4. Descripción videos Virat

Video	Descripción
3_2012-07-17_19-46-13	No fuego No humo
3_2012-07-17_20-16-17	No fuego No humo

3_2012-07-18_19-56-15_1	No fuego No humo
TGAVideo	Si fuego No humo
VIRAT_S_000002_sizesmall	No fuego No humo
VIRAT_S_010005_04_000299_000323	No fuego No humo

3.3.3 Videos de fuego de Youtube

Adicionalmente, se realizó una selección de videos libres desde la plataforma de YouTube donde había presencia de fuego/humo. La Tabla 5 presenta la descripción y enlaces para los 13 videos usados desde la plataforma youtube.

Tabla 5. Videos de flama seleccionados de la plataforma libre youtube

Nombre video	Descripción	Link
fire blue	Fuego creado con fluidos mayas con fondo negro	http://youtube.com/watch?v=V1_4BNN4in8
Sony Vegas Blue FIRE Effect	Efecto de fuego azul con fondo negro	http://youtube.com/watch?v=Hf-CQUaAeN4
Free Movie Channel	Llamas azules de una quema de gas natural y la reflexión sobre fondo negro	http://youtube.com/watch?v=FarwVPa74XQ
Blue Flames	Llamas azules continuas en fondo negro	http://youtube.com/watch?v=fqMAHNWBHdk
Free Stock Footage HD Flames	Explosión fuego amarillo en 1080 pixel fondo negro	http://youtube.com/watch?v=HgKd3FxSSw4
Fire 50 fps Footage	Secuencias de video azul con fondo negro	http://youtube.com/watch?v=JZakqoTtrZM
Fractal Noise Fire in After Effects	Fuego amarillo en fondo negro	http://youtube.com/watch?v=g5RQmJtmxVA
Slow Motion Fire in Blender	Fuego explosión lento fondo negro	http://youtube.com/watch?v=IRFrAfyilj4
BG Fire Black Screen Effect	Fuego amarillo en HD fondo negro	http://youtube.com/watch?v=XGnnYiwK4Dw
Fire Animation Background-HD	Fuego animación amarillo fondo negro	http://youtube.com/watch?v=0G9rm7Jen3c
Slow Motion Fire HD Motion Background	Fuego amarillo lateral fondo negro	http://youtube.com/watch?v=9Rnq0XtgadY

Low flames flickering with a black background.	Bajas llamas parpadeo con un fondo negro.	http://youtube.com/watch?v=Tt6PBKHo-oQ
Slow Motion Fire Blaze From the Bottom Stock Video Footage	Fuego amarillo fondo negro	http://youtube.com/watch?v=cPYaQ- MKt0

3.2 Metodología propuesta

Para la detección automática de eventos relacionados con incendios en entornos abiertos, se utilizaron los videos en secuencias de imágenes (frame). Posteriormente estos frame se segmentaron con las técnicas Background Subtraction usando las funciones de Matlab Diff (diferencia absoluta entre dos frames) y con la BitXor (Operador Binario Xor entre dos frames) de detección de movimiento. Para realizar esta segmentación se empleó la topología en cascada y en paralelo con 7 espacios de color RGB, YCbCr, CMY, CMYK, HSI, HSL Y HSV. Una vez segmentadas las imágenes se realizó una extracción de 14 características seleccionadas de acuerdo al estado del arte. Con estos datos, se realiza la comparación de las técnicas de segmentación, para realizar la clasificación y entrenamiento.

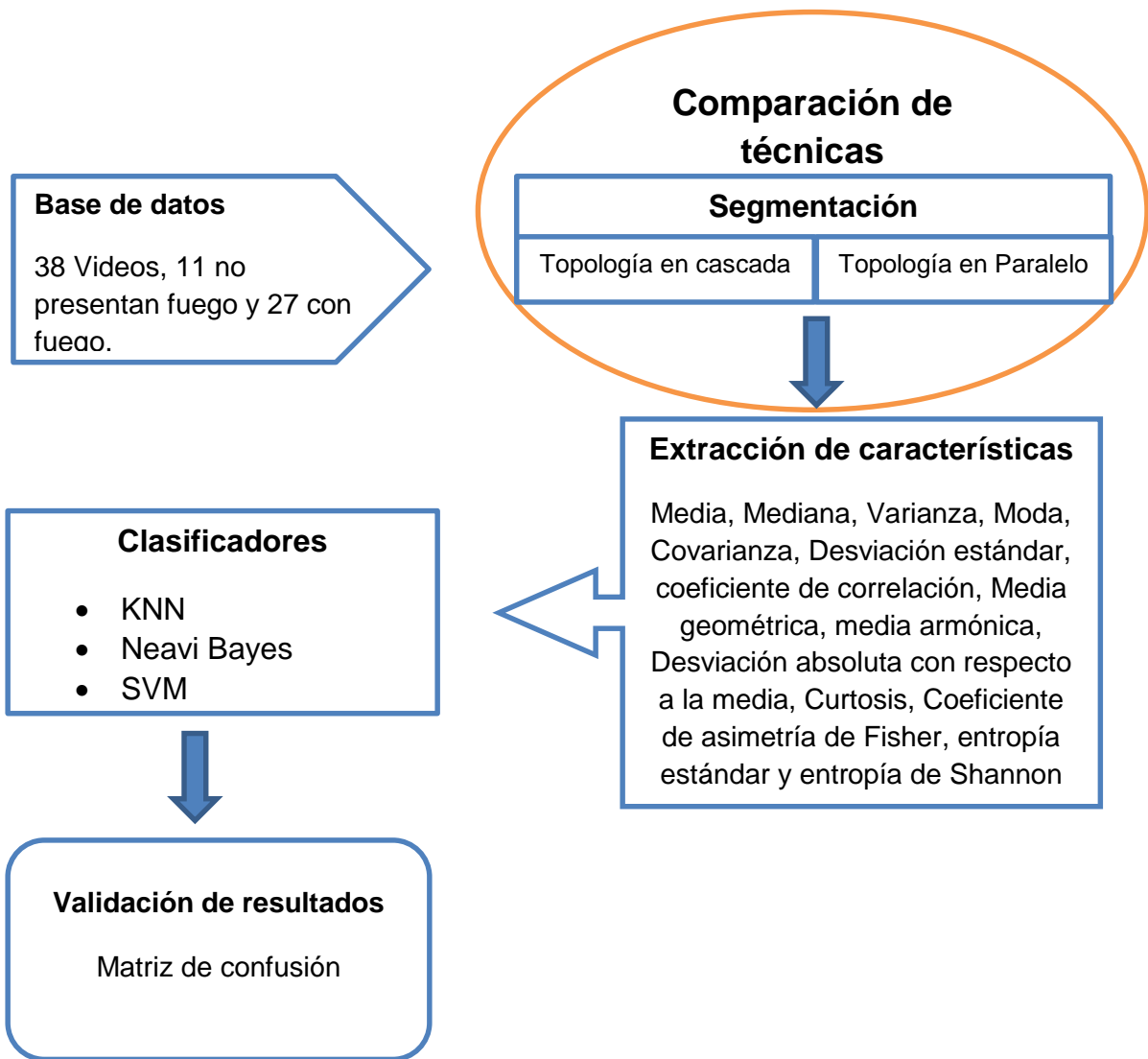
Para la selección de las características, se tuvo en cuenta que los métodos que utilizan el color se basan en la consideración de que una llama, bajo la suposición de que es generada por combustibles comunes como madera, plástico, papel u otro, se puede caracterizar de manera confiable por su color ya que aportan información relevante y propiedades medibles. En este paso se transforma la imagen en un nuevo espacio de representación, obteniendo características estadísticas, de textura u otros con el fin de mejorar la representación de los objetivos y componentes de la imagen.

Con base a lo anterior se seleccionaron características estadísticas que aportaran información básica e información relevante y que con se pudiera desarrollar la metodología propuesta para la detección de fuego.

En la extracción de características se creó una matriz de características a partir de las imágenes originales de la base de datos y las imágenes segmentadas, por cada capa de

color. A la matriz resultado se le realiza un análisis de relevancia donde se determinó que características no presentaban variación de datos, (se busca que los datos varíen), finalmente con la matriz de características se entrena con los métodos de clasificadores KNN, NAYVE BAYES y SVM y se validan los datos con la matriz de confusión para calcular la precisión, como se muestra en la figura siguiente.

Figura 2. Metodología en video fire detection propuesta.



3.3 Comparacion de técnicas

La comparación de técnicas de segmentación se basó en separar los píxeles del fuego en los videos de la base de datos usando dos topologías, una en cascada y la otra en paralelo. A partir de la segmentación se obtuvo una tabla comparativa para evaluar cuál es la topología que mejor resultados da y una matriz de datos de los videos segmentados, a partir de estos datos se realizó la extracción de características.

En la segmentación se trabajó de forma independiente con la detección de movimiento y la segmentación por color. A partir de los experimentos se encontró que individualmente se presentaba mucho ruido en la segmentación de las imágenes (frames), para contrarrestar el ruido se reunieron los resultados parciales de ambas con un operador lógico and, dando como resultado una topología de conexión de bloques en paralelo. También se ensayó una topología de conexión de bloques en serie o cascada para crear un marco comparativo entre ambas topologías y determinar cuál de estas presenta mejor respuesta a la segmentación de la flama.

Es de anotar que el ajuste de parámetros del fuego en los espacios de color RGB y YCbCr, son tomados de la literatura, el resto de los parámetros se ajustaron manualmente durante las pruebas (Binti Zaidi, Binti Lokman, Bin Daud, Achmad, & Chia, 2015)..

Tabla 6. Reglas de espacio de color

Color space	Rules
RGB	1) $R > G > B$
	2) $\text{if } R > R_{\text{mean}} \cap G > G_{\text{mean}} \cap B < B_{\text{mean}}$
YCbCr	3) $Y(x,y) \geq Cb(x,y)$
	4) $Cr(x,y) \geq Cb(x,y)$
	5) $Y(x,y) \geq Y_{\text{mean}} \cap Cb(x,y) \leq Cb_{\text{mean}} \cap Cr(x,y) \geq Cr_{\text{mean}}$
	6) $Cb(x,y) - Cr(x,y) \geq Th$
	7) $(Cb(x,y) \leq 120) \cap (Cr(x,y) \geq 150)$

Espacio de Color CMY

$$C < M < Y$$

$$C \leq C_{\text{mean}} \cap M \leq M_{\text{mean}} \cap Y \geq Y_{\text{mean}}$$

Espacio de Color CMYK

$$C < M < Y$$

$$C \leq C_{mean} \cap M > M_{mean} \cap Y \geq Y_{mean}$$

Espacio de Color HSV

$$Hue < Hue_{mean} \cap Saturation > Saturation_{mean} \cap Value > Value_{mean}$$

Espacio de Color HSL

$$Hue < Hue_{mean} \cap Saturation > \frac{Saturation_{mean}}{2} \cap Lightness > Lightness_{mean}$$

Espacio de Color HSI

$$Hue < Hue_{mean} \cap Saturation > Saturation_{mean} \cap Intensity > Intensity_{mean}$$

3.3.1 Topología en Cascada

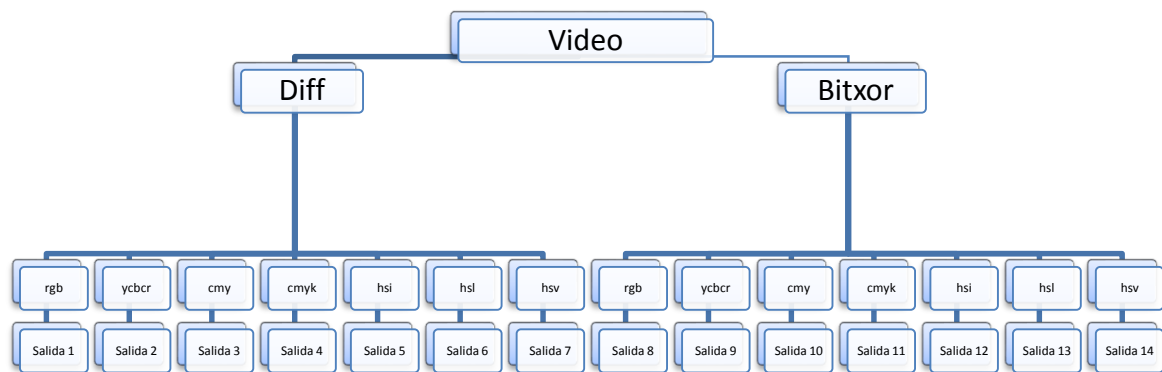
La topología en cascada es una técnica lineal para segmentar videos(D. Zhang & Wang, 2016). Primero se realiza a detección de movimiento con las técnicas background subtraction Diff como con la BitXor operadas en Matlab sobre los videos de fuego, posteriormente se obtienen por separado los resultados obtenidos a partir de cada una de estas técnicas.

Se toma el resultado que dio la técnica de movimiento Diff y este el resultado obtenido se le segmenta a partir de los espacios de color RGB, YCbCr, CMY, CMYK, HSI, HSL y HSV, obteniendo así 7 resultados de la segmentación por cada espacio de color.

De igual manera se toma el resultado que dio la técnica de movimiento Bitxor y este el resultado obtenido se le segmenta a partir de los espacios de color RGB, YCbCr, CMY, CMYK, HSI, HSL y HSV, obteniendo así 7 resultados de la segmentación por cada espacio de color.

De la topología de cascada se obtienen 14 resultados (7 de la técnica de movimiento diff y 7 de la técnica de movimiento Bitxor) para realizar la evaluación comparativa entre las topologías cascada y paralelo, y las matrices de datos para la extracción de características. En la siguiente figura se muestra el desarrollo de la topología en cascada.

Figura 3. Topología en Cascada.



3.3.2 Topología en Paralelo

Es una técnica para segmentar de manera simultánea e independientemente las técnicas de movimiento y los espacios de color y posteriormente combina los resultados por medio de un multiplicador lógico (D. Zhang & Wang, 2016)(and).

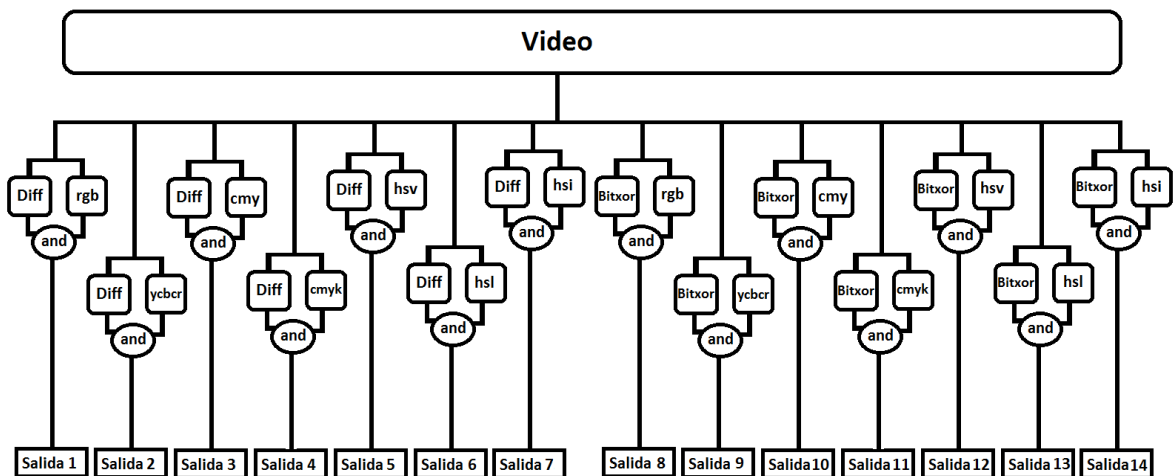
Se toma el video de la base de datos se le aplica la técnica de movimiento Diff, paralelamente se toma el mismo video de la base de datos y se segmenta a partir de los 7 espacios de color RGB, YCBCR, CMY, CMYK, HSI, HSL y HSV, posteriormente se combina el resultado de la técnica de movimiento Diff con cada uno de los resultados de

la segmentación de los espacios de color por medio de una and, obteniendo así 7 resultados de la segmentación.

De igual forma se toma el video de la base de datos se le aplica la técnica de movimiento Bitxor, paralelamente se toma el mismo video de la base de datos y se segmenta a partir de los 7 espacios de color RGB, YCBCR, CMY, CMYK, HSI, HSL y HSV, posteriormente combina el resultados de la técnica de movimiento Bitxor con cada uno de los resultados de la segmentación de los espacios de color por medio de una And, obteniendo así 7 resultados de la segmentación.

De esta topología en paralelo se obtienen 14 resultados (7 de la técnica de movimiento diff y 7 de la técnica de movimiento Bitxor) para realizar la evaluación comparativa entre las topologías cascada y paralelo, y las matrices de datos para la extracción de características. En la siguiente figura se muestra el desarrollo de la topología en paralelo.

Figura 4. Topología en Paralelo.



3.4 Extracción de características

En este paso se transforma la imagen en un nuevo espacio de representación, obteniendo características estadísticas, de textura u otros con el fin de mejorar la representación de los objetivos y componentes de la imagen.

A partir de las matrices de la segmentación se crea una matriz de características a través de la media, mediana, varianza, moda, covarianza, desviación estándar, coeficiente de correlación, media geométrica, media armónica, desviación absoluta con respecto a la media, curtosis, coeficiente de asimetría de Fisher, entropía estándar y entropía de Shannon, respectivamente.

En algunos artículos se tomaron la desviación estándar como característica, se toma la entropía para tratar de establecer alguna característica respecto a la energía debido a la degradación de energía mencionada en varios papers y los coeficientes de Fisher para establecer características referentes a la simetría de la región segmentada.

Esta matriz de características se obtiene de 14 características sobre cada espacio de color de los videos originales y 14 características sobre cada capa en un espacio de color en los videos segmentados, como resultado cada una de las características entrega 3 resultados (uno por cada capa en un espacio de color) con un total de 84 valores (catorce características de video original mas catorce características de video segmentado esto multiplicado por las tres capas de color) estas características aplicadas a 2759 frame, se genera finalmente una matriz características de 84×2759 . A partir de la matriz característica se entrenan los clasificadores.

3.5 Clasificación

Desde el nuevo espacio de representación, se clasifican las secuencias de imágenes usando las técnicas KNN, Nevi Bayes y SVM.

Un clasificador recibe una descripción de un objeto para ser catalogado, decide la clase de objetos a la que pertenece y entrega como salida la etiqueta de la clase apropiada. Para un sistema de decisión, cada registro contiene un conjunto de atributos donde uno de los

atributos es la clase. Encontrar un modelo para el atributo de la clase como una función de los valores de los otros atributos, es la tarea fundamental de un clasificador, para ello normalmente la base de datos es dividida en dos subconjuntos, uno de entrenamiento que se utiliza para construir el modelo y uno de prueba para determinar la precisión del modelo o validarlo. Una regla común para dividir la base de datos es utilizar el 90% para entrenamiento y el 10% para validación, sin embargo, esta división es usualmente aleatoria, con el fin de garantizar que ambos subconjuntos son muestras aleatorias de la misma distribución, también puede ser razonable hacer un muestreo estratificado, lo que significa asegurarse de que cada clase está presente en la misma proporción en los subconjuntos de entrenamiento y validación (Becerra Botero, 2013).

Este entrenamiento y validación cruzada se realiza diez veces nunca repitiendo el 90% de conjunto de datos ni el 10% de conjunto de datos, siempre se realiza aleatorio, Garantizando así que se entrena y valida con el total de los datos de la matriz característica.

Las redes Bayesianas (Naive Bayes) son consideradas como un método óptimo en la tarea de clasificación, no sólo porque ofrecen un análisis cualitativo de los datos mediante la correlación entre sus atributos, sino también porque dan cuenta de la importancia cuantitativa de esos atributos mediante una medida probabilística, lo que hace la diferencia fundamental de las redes Bayesianas respecto a otros métodos, como árboles de decisión y redes neuronales las cuales no proporcionan una medida cuantitativa de la clasificación. Usar el teorema de Bayes en cualquier problema de aprendizaje automático (en especial los de clasificación) permite estimar las probabilidades a posteriori de cualquier hipótesis consistente con el conjunto de datos de entrenamiento, para así escoger la hipótesis más probable; para estimar estas probabilidades se han propuesto numerosos algoritmos, entre los que cabe destacar el algoritmo Naive Bayes (Becerra Botero, 2013).

Para evaluar el desempeño del clasificador con cualquier entrada se puede emplear una métrica de precisión y el llamado de pixel como en (T. Ko et al., 2008), definidos como se presenta a continuación:

$$\text{Precisión: } \frac{\# \text{ verdaderos positivos}}{\# \text{ verdaderos positivos} + \# \text{ falsos positivos}}$$

$$\text{Rellamado: } \frac{\# \text{ verdaderos positivos}}{\# \text{ verdaderos positivos} + \# \text{ falsos Negativos}}$$

4. Resultados

Este capítulo presenta los resultados de la comparación de las topologías en cascada y paralelo para la segmentación de flama/humo en secuencia de imágenes, así como las técnicas de extracción de características y clasificación.

4.1 Comparacion de tecnicas

La Tabla 7 y 8 presenta los porcentajes de detección de pixeles que representan fuego en los 12 videos analizados. Las tablas de los resultados fueron diseñadas en tres bloques de columnas: la primera columna identifica el video analizado; la segunda columna presenta el numero de pixeles por frame; y en la tercera, los siete espacios de color a partir de los cuales se aplico las topologías de segmentación en cascada y paralelo. En las tablas se resalta el mejor resultado para cada video. Los resultados se presentan gráficamente en la Figuras 5 y 6, donde cada línea de color representa un espacio de color y el eje Y presenta los 12 videos analizados.

Tabla 7. Porcentaje de detección de la topología de segmentación en cascada

Video	Píxeles x Frame	Espacio de color (%)						
		rgb	ycbcr	cmy	cmyk	hsv	hsl	hsi
BG_Fire_Black_Screen_Effect	920986,6	4,19	0,00	1,02	0,00	0,57	1,00	1,04
Blue_Flames	362762,0	21,01	0,00	0,06	0,00	0,03	0,25	0,19
Blue_flame_loop	230399,8	9,32	0,00	0,00	0,00	0,00	0,00	0,00
Fractal_Noise_Fire_in_After_Effects_No_3rd_party_p	542729,3	34,81	0,00	7,47	0,09	2,48	6,03	7,02
Free_Stock_Footage	708013,0	20,07	0,00	2,71	0,03	2,09	4,67	4,83
Low_flames_flickering_with_a_black_background	70240,2	105,15	0,01	9,83	0,02	10,48	12,12	14,09
Realistic_CG_Fire	103231,1	61,84	0,01	6,44	0,07	7,40	7,96	9,57
Slow_Motion_Fire_Blaze_From_the_Bottom_Stock_Video	921598,2	3,65	0,01	0,09	0,00	0,15	0,20	0,22
Slow_Motion_Fire_HD_Motion_Background	921599,8	0,70	0,00	0,01	0,00	0,02	0,03	0,04
Slow_Motion_Fire_in_Blender	180068,9	6,46	0,00	1,58	0,00	0,87	0,95	0,87
Sony_Vegas_Blue_FIRE_Effect	6912,0	43,44	0,02	0,06	0,00	0,48	0,03	0,02
TGAVideo	27327,8	57,50	0,00	7,96	0,79	11,57	9,31	13,67
fire_blue	29329,0	30,25	0,00	0,15	0,01	0,22	0,04	0,00

En la Tabla 7 se evidencia que los mejores porcentajes de detección de fuego usando la topología de segmentación en cascada se obtiene a partir del espacio de color RGB. Por ejemplo, para el video Realistic_CG_Fire se obtiene un 61.84% de precisión, para el video Sony_Vegas_Blue_FIRE_Effect un 43.44% y TGAVideo un 57.50% a partir del espacio RGB. Para el video Low_flames_flickering_with_a_black_background en este mismo espacio de color se obtuvo un porcentaje de 105.15% lo que indica que hubo una falsa detección de fuego. Para otros videos el porcentaje de detección fue menor al 10% permitiendo deducir que la topología en cascada no aporta significativamente a la detección.

Figura 5. Resultados de la topología de segmentación en cascada.

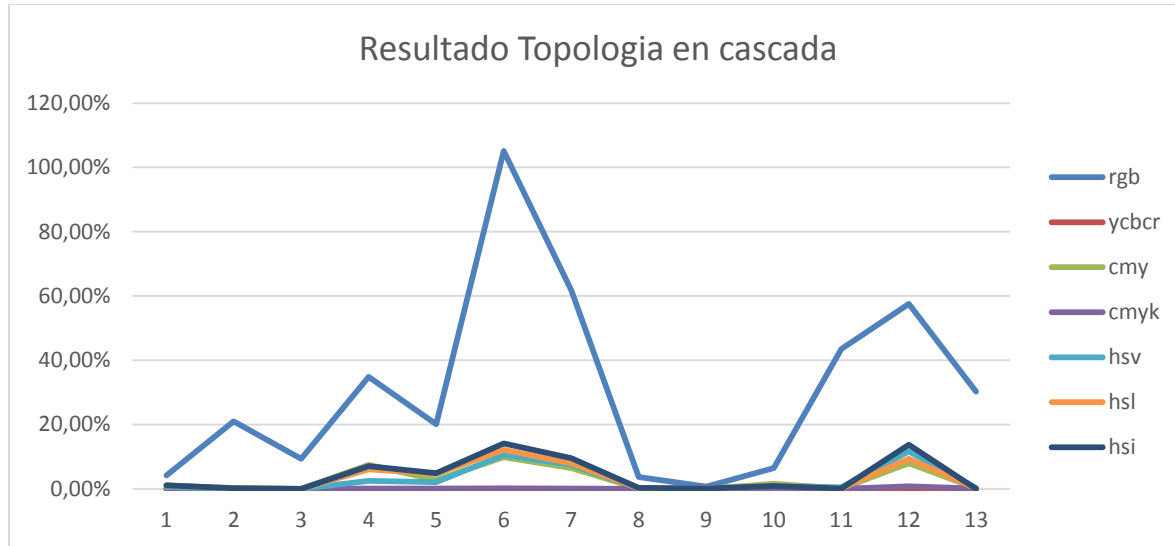
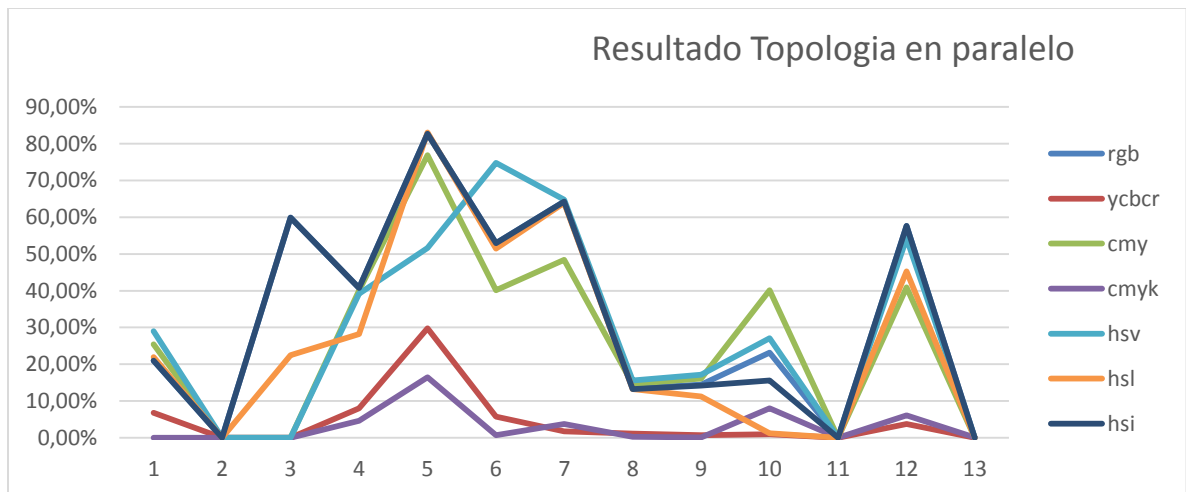


Tabla 8. Resultados de la topología de segmentación en paralelo

Video	Píxeles x Frame	Espacio de color (%)						
		rgb	ycbcr	cmy	cmyk	hsv	hsl	hsi
BG_Fire_Black_Screen_Effect	920986,565	20,98	6,76	25,43	0,00	28,97	21,91	20,98
Blue_Flames	362761,9697	0,00	0,00	0,00	0,04	0,00	0,00	0,00
Blue_flame_loop	230399,7895	59,91	0,00	0,00	0,00	0,00	22,47	59,91
Fractal_Noise_Fire_in_After_Effects_No_3rd_party_p	542729,3191	40,78	8,06	40,07	4,63	39,20	28,22	40,78
Free_Stock_Footage	708012,9730	82,70	29,81	76,82	16,47	51,64	83,08	82,69
Low_flames_flickering_with_a_black_background	70240,1500	52,94	5,73	40,13	0,72	74,81	51,49	52,94
Realistic_CG_Fire	103231,0909	64,26	1,76	48,42	3,73	64,76	63,92	64,26
Slow_Motion_Fire_Blaze_From_the_Bottom_Stock_Video	921598,2121	13,55	1,14	14,63	0,30	15,58	13,23	13,27
Slow_Motion_Fire_HD_Motion_Background	921599,7677	14,46	0,73	16,16	0,07	17,17	11,26	14,21
Slow_Motion_Fire_in_Blender	180068,8857	23,17	1,01	40,14	8,07	27,11	1,23	15,61
Sony_Vegas_Blue_FIRE_Effect	6912,0000	0,00	0,02	0,00	0,01	0,03	0,00	0,00
TGAVideo	27327,8000	57,60	3,74	40,93	6,10	54,23	45,29	57,60
fire_blue	29329,0256	0,00	0,00	0,00	0,12	0,00	0,00	0,00

Los resultados de la segmentación bajo la topología en paralelo presentan buenos resultados usando los modelos de color rgb, hsv, hsl y hsi. Por ejemplo, para los videos Blue_flame_loop y Fractal_Noise_Fire_in_After_Effects_No_3rd_party_p los mejores resultados de detección fueron del 59.91% y 40.78% respectivamente usando los espacios de color RGB y HSI. Por su parte, para el video Free_Stock_Footage el mejor resultado se obtuvo con el espacio HSL con un porcentaje de detección de 83.08%. Usando el modelo hsv se obtiene los más altos porcentajes de detección para los videos Low_flames_flickering_with_a_black_background y Realistic_CG_Fire. Los videos con fuego amarillo fueron detectados generalmente por todos los espacios, sin embargo, los videos de fuego azul no fueron segmentados por ningún espacio de color. Con estos resultados, se estima que la topología en paralelo es una técnica prometedora para la segmentación y detección de fuego.

Figura 6. Resultados de la topología de segmentación en paralelo.



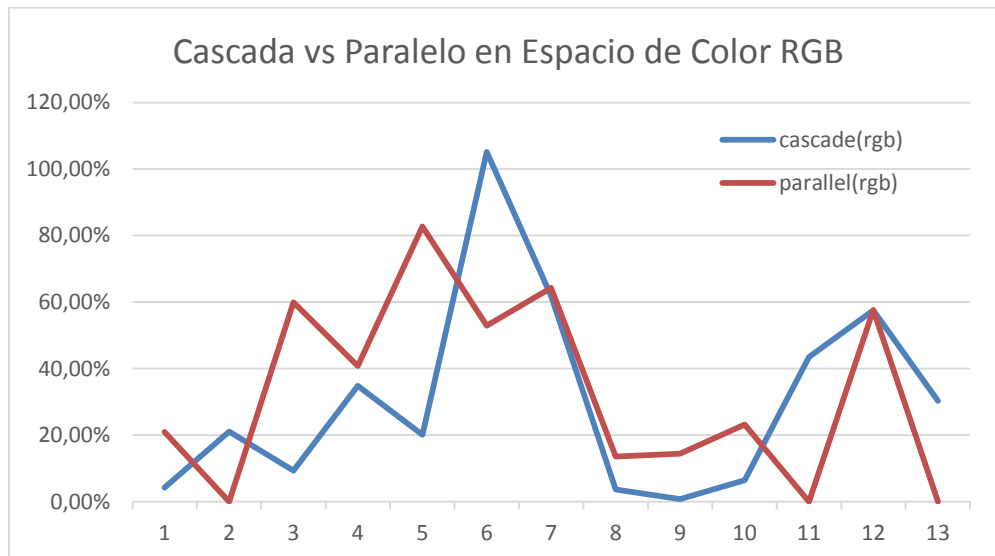
Se presentan en las siguientes tablas y figuras el resultado de la evaluación de cada espacio de color respecto a cada topología, finalmente se describe de forma sucinta los datos o características más relevantes de los resultados por espacio de color.

Para el espacio de color RGB se presenta la comparación de las topologías en cascada y paralelo en la Tabla 9 y Figura 7.

Tabla 9. Resultados topología cascada vs topología en paralelo en espacio de color RGB

Video	Pixeles x Frame	Cascada (%)	Paralelo (%)
BG_Fire_Black_Screen_Effect	920986,57	4,19	20,98
Blue_Flames	362761,97	21,01	0,00
Blue_flame_loop	230399,79	9,32	59,91
Fractal_Noise_Fire_in_After_Effects_No_3rd_party_p	542729,32	34,81	40,78
Free_Stock_Footage	708012,97	20,07	82,70
Low_flames_flickering_with_a_black_background	70240,15	105,15	52,94
Realistic_CG_Fire	103231,09	61,84	64,26
Slow_Motion_Fire_Blaze_From_the_Bottom_Stock_Video	921598,21	3,65	13,55
Slow_Motion_Fire_HD_Motion_Background	921599,77	0,70	14,46
Slow_Motion_Fire_in_Blender	180068,89	6,46	23,17
Sony_Vegas_Blue_FIRE_Effect	6912,00	43,44	0,00
TGAVideo	27327,80	57,50	57,60
fire_blue	29329,03	30,25	0,00

Figura 7. Resultados topología cascada vs topología en paralelo en espacio de color RGB



De los resultados en la Tabla 9 se puede observar que los mejores resultados de segmentación de las regiones con presencia de fuego se obtienen con con la topología en paralelo dada específicamente en la imagen de fuego amarillo con un 82,70% teniendo en cuenta que las flamas de color azul, no fueron detectadas en esta misma topología. Se

observa además que el resultado de la comparación, el video Low flames flickering with a black background en la topología en paralelo hubo un valor por encima del 100% (105,15%) lo que significa un falso positivo.

Las Figuras 8 y 9 presentan la secuencia de imagen de fuego con fondo negro tomadas después de haber aplicado las topologías en cascada y en paralelo en espacio de color RGB, respectivamente. En estas se puede observar que la segmentación en cascada de esta secuencia de imágenes fue mejor que en paralelo.

Para el espacio de color YCBCR la comparación se presenta en la Tabla 10 y Figura 10 con los resultados de la comparación de la topología cascada vs topología en paralelo.

Figura 8. Topología cascada en espacio de color RGB



Figura 9. Topología paralelo en espacio de color RGB

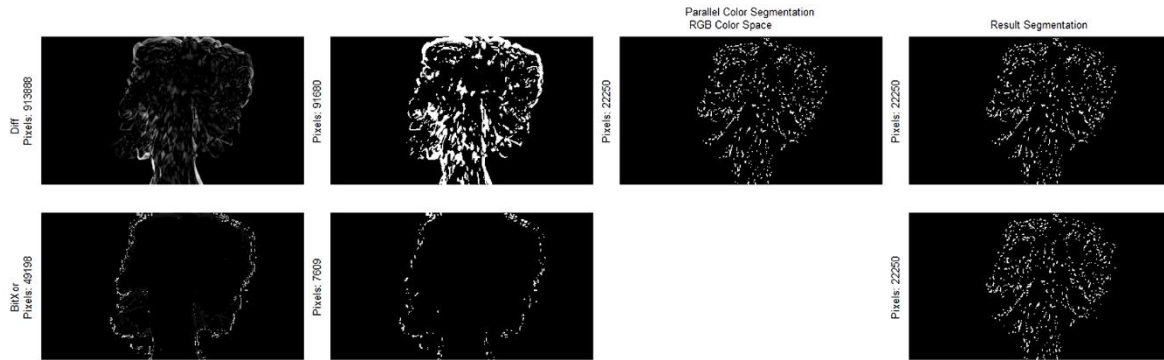
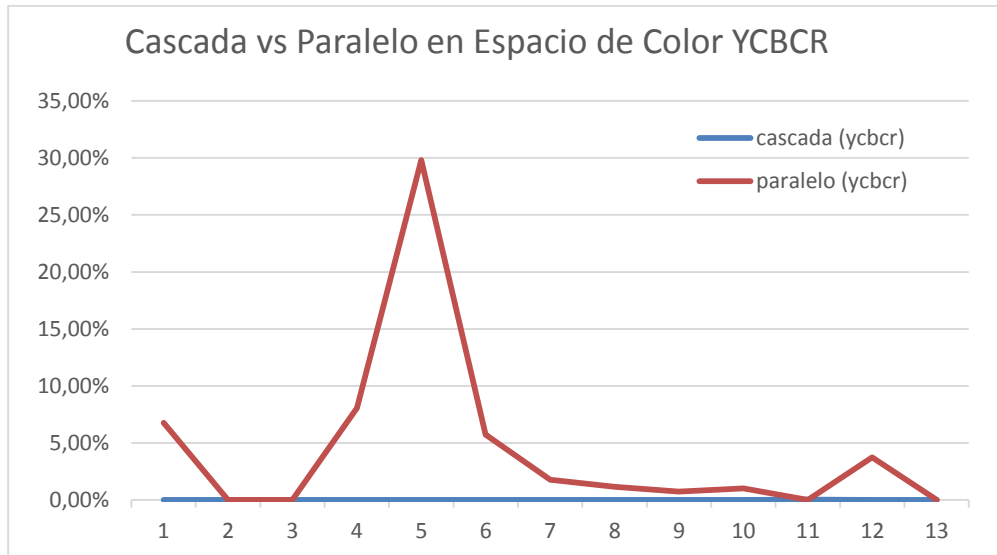


Tabla 10. Resultados topología cascada vs topología en paralelo en espacio de color YCBCR.

Video	Pixeles x Frame	cascada (%)	paralelo (%)
BG_Fire_Black_Screen_Effect	920986,57	0,00	6,76
Blue_Flames	362761,97	0,00	0,00
Blue_flame_loop	230399,79	0,00	0,00
Fractal_Noise_Fire_in_After_Effects_No_3rd_party_p	542729,32	0,00	8,06
Free_Stock_Footage	708012,97	0,00	29,81
Low_flames_flickering_with_a_black_background	70240,15	0,01	5,73
Realistic_CG_Fire	103231,09	0,01	1,76
Slow_Motion_Fire_Blaze_From_the_Bottom_Stock_Video	921598,21	0,01	1,14
Slow_Motion_Fire_HD_Motion_Background	921599,77	0,00	0,73
Slow_Motion_Fire_in_Blender	180068,89	0,00	1,01
Sony_Vegas_Blue_FIRE_Effect	6912,00	0,02	0,02
TGAVideo	27327,80	0,00	3,74
fire_blue	29329,03	0,00	0,00

Resultados topología cascada vs topología en paralelo en espacio de color YCBCR

Figura 10.



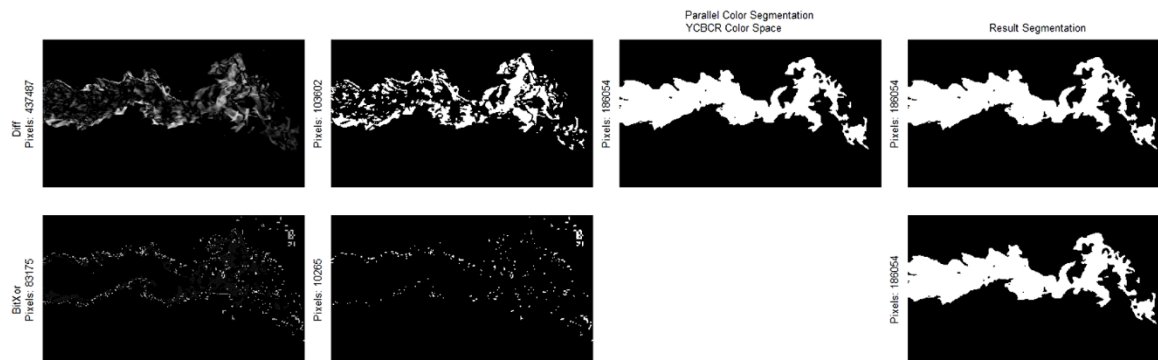
De la Tabla 10 y Figura 10 respectivamente, se puede observar de forma general, que los resultados del espacio de color YCbCr en la metodología en paralelo lograron segmentar algunas imágenes mientras que los de cascada, no. Se puede ver que los videos de flama azul afectan directamente los resultados en ambas metodologías en la detección de la flama.

Las Figuras 11 y 12, corresponden a la secuencia de imagen de fuego con fondo negro tomadas después de haber aplicado las topologías en cascada y en paralelo en espacio de color YCBCR, respectivamente.

Figura 11. Topología cascada en espacio de color YCBCR



Figura 12. Topología paralelo en espacio de color YCBCR



Para el espacio de color CMY se presenta la comparación de los resultados de segmentación en la Tabla 11 y Figura 13 con la topología cascada vs topología en paralelo.

En la Tabla 11 y Figura 13 se logra ver que el porcentaje de detección de imágenes de fuego con el espacio de color CMY en la metodología en paralelo son superiores que con la metodología en cascada. Los videos de flama azul inciden en resultados incipientes de detección para ambas tecnicas y finalmente se destaca que en el espacio CMY solo se logro un 10%. Las Figuras 14 y 15, permiten observar algunos resultados de las topologías en cascada y paralelo en el espacio de color CMY.

Tabla 11. Resultados topología cascada vs topología en paralelo en espacio de color CMY

Video	Pixeles x Frame	cascada (%)	paralelo (%)
BG_Fire_Black_Screen_Effect	920986,57	1,02	25,43
Blue_Flames	362761,97	0,06	0,00
Blue_flame_loop	230399,79	0,00	0,00
Fractal_Noise_Fire_in_After_Effects_No_3rd_party_p	542729,32	7,47	40,07
Free_Stock_Footage	708012,97	2,71	76,82
Low_flames_flickering_with_a_black_background	70240,15	9,83	40,13
Realistic_CG_Fire	103231,09	6,44	48,42
Slow_Motion_Fire_Blaze_From_the_Bottom_Stock_Video	921598,21	0,09	14,63
Slow_Motion_Fire_HD_Motion_Background	921599,77	0,01	16,16
Slow_Motion_Fire_in_Blender	180068,89	1,58	40,14
Sony_Vegas_Blue_FIRE_Effect	6912,00	0,06	0,00
TGAVideo	27327,80	7,96	40,93
fire_blue	29329,03	0,15	0,00

Figura 13. Resultados topología cascada vs topología en paralelo en espacio de color CMY

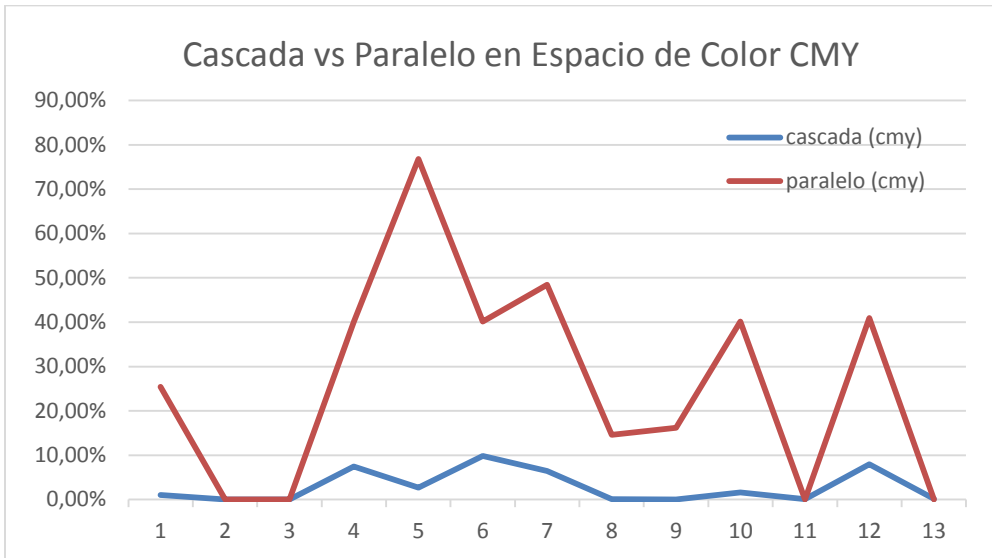


Figura 14. Topología cascada en espacio de color CMY

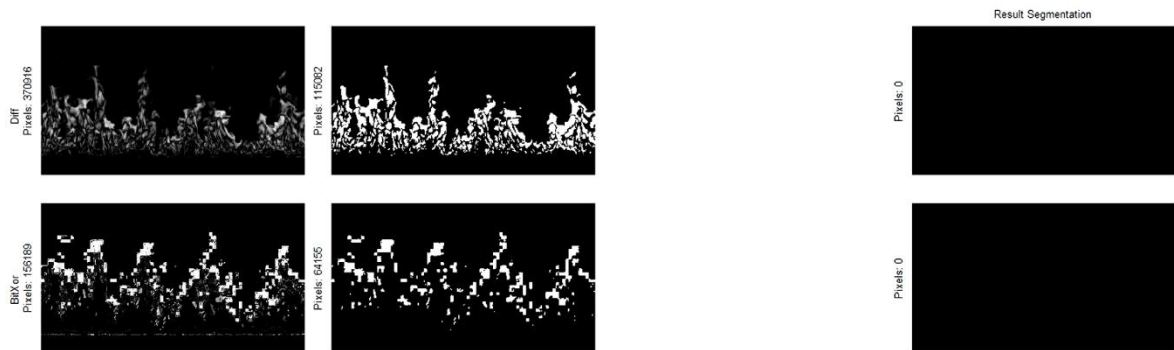
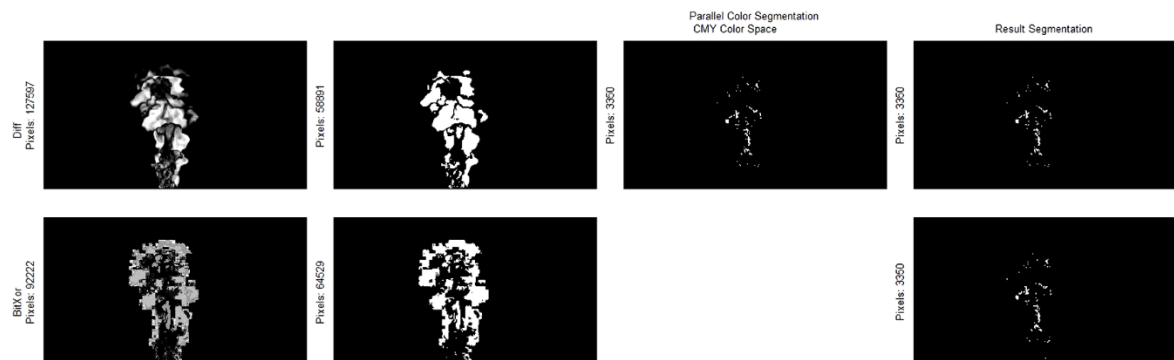


Figura 15. Topología en paralelo en espacio de color CMY

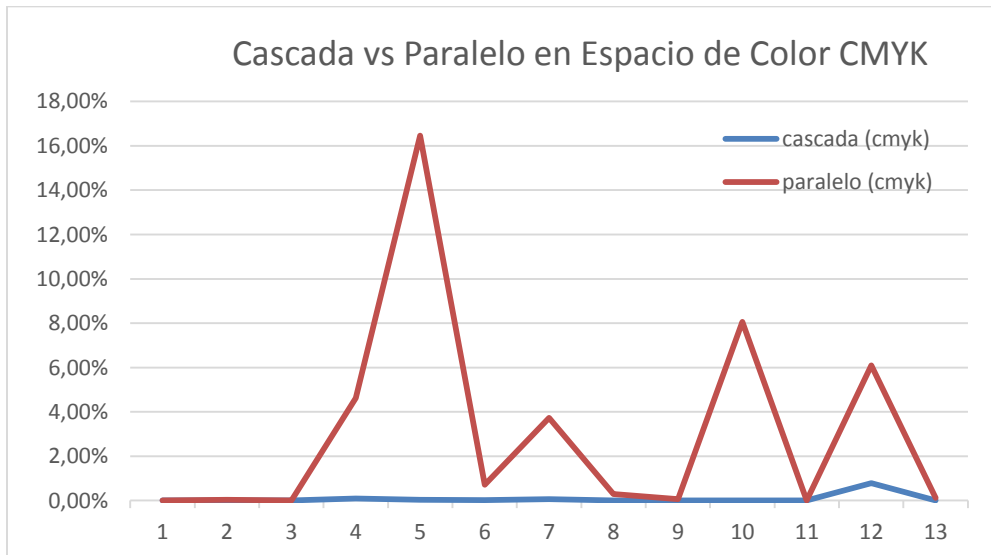


La Tabla 12 y Figura 16 presentan los datos de la comparación usando la metodología en paralelo Vs la metodología en cascada respectivamente en el espacio de color CMYK.

Tabla 12. Resultados topología cascada vs topología en paralelo en espacio de color CMYK

Video	Pixeles x Frame	cascada (%)	paralelo (%)
BG_Fire_Black_Screen_Effect	920986,57	0,00	0,00
Blue_Flames	362761,97	0,00	0,04
Blue_flame_loop	230399,79	0,00	0,00
Fractal_Noise_Fire_in_After_Effects_No_3rd_party_p	542729,32	0,09	4,63
Free_Stock_Footage	708012,97	0,03	16,47
Low_flames_flickering_with_a_black_background	70240,15	0,02	0,72
Realistic_CG_Fire	103231,09	0,07	3,73
Slow_Motion_Fire_Blaze_From_the_Bottom_Stock_Video	921598,21	0,00	0,30
Slow_Motion_Fire_HD_Motion_Background	921599,77	0,00	0,07
Slow_Motion_Fire_in_Blender	180068,89	0,00	8,07
Sony_Vegas_Blue_FIRE_Effect	6912,00	0,00	0,01
TGAVideo	27327,80	0,79	6,10
fire_blue	29329,03	0,01	0,12

Figura 16. Resultados topología cascada vs topología en paralelo en espacio de color CMYK



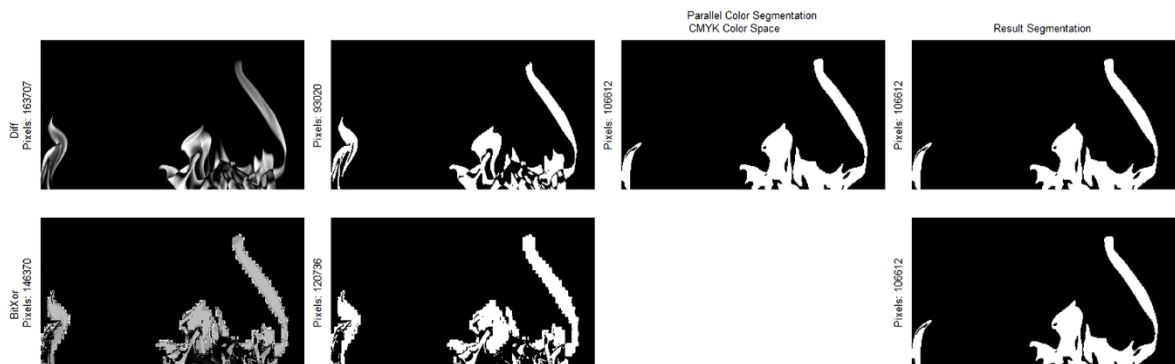
En la Tabla 12 y Figura se puede observar que los resultados del espacio de color CMY en la metodología en paralelo alcanzan una detección del 20% mientras que en cascada no supera el 1%. De igual manera que en los anteriores espacios de color, CMY no detecta imágenes de flama azul.

Las Figuras 16, 17 y 18 permite visualizar la imagen de referencia de los resultados de la metodología en cascada vs topología en paralelo en el espacio de color CMYK y además, observar algunos secuencias de imágenes complementarias a los resultados de las topologías en cascada y paralelo en el espacio de color CMYK.

Figura 17. Topología cascada en espacio de color CMYK



Figura 18. Topología paralelo en espacio de color CMYK

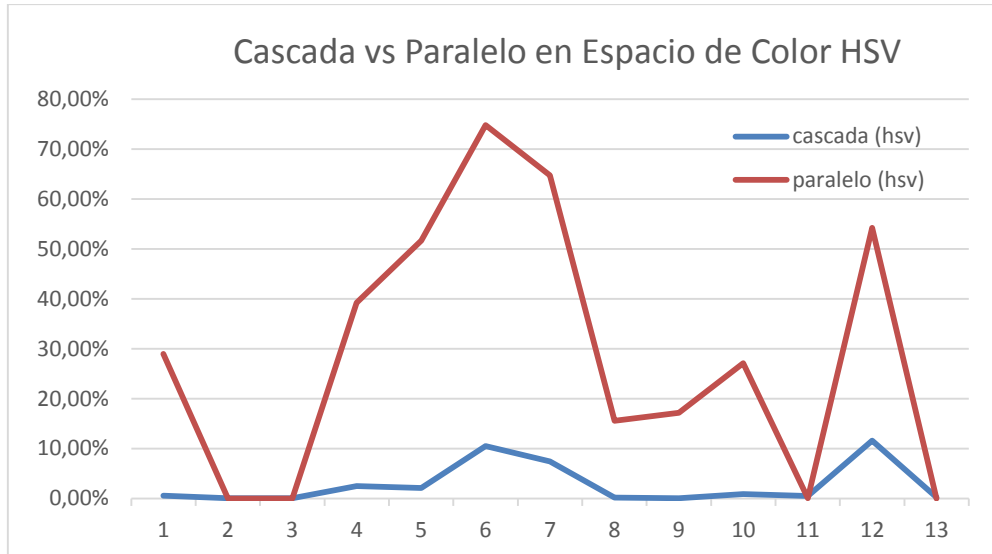


Para el espacio de color HSV, la Tabla 13 y Figura 19 presenta los resultados de la comparación de la topología cascada vs topología en paralelo

Tabla 13. Resultados topología cascada vs topología en paralelo en espacio de color HSV

Video	Píxeles x Frame	cascada (%)	paralelo (%)
BG_Fire_Black_Screen_Effect	920986,57	0,57	28,97
Blue_Flames	362761,97	0,03	0,00
Blue_flame_loop	230399,79	0,00	0,00
Fractal_Noise_Fire_in_After_Effects_No_3rd_party_p	542729,32	2,48	39,20
Free_Stock_Footage	708012,97	2,09	51,64
Low_flames_flickering_with_a_black_background	70240,15	10,48	74,81
Realistic_CG_Fire	103231,09	7,40	64,76
Slow_Motion_Fire_Blaze_From_the_Bottom_Stock_Video	921598,21	0,15	15,58
Slow_Motion_Fire_HD_Motion_Background	921599,77	0,02	17,17
Slow_Motion_Fire_in_Blender	180068,89	0,87	27,11
Sony_Vegas_Blue_FIRE_Effect	6912,00	0,48	0,03
TGAVideo	27327,80	11,57	54,23
fire_blue	29329,03	0,22	0,00

Figura 19. Resultados topología cascada vs topología en paralelo en espacio de color HSV



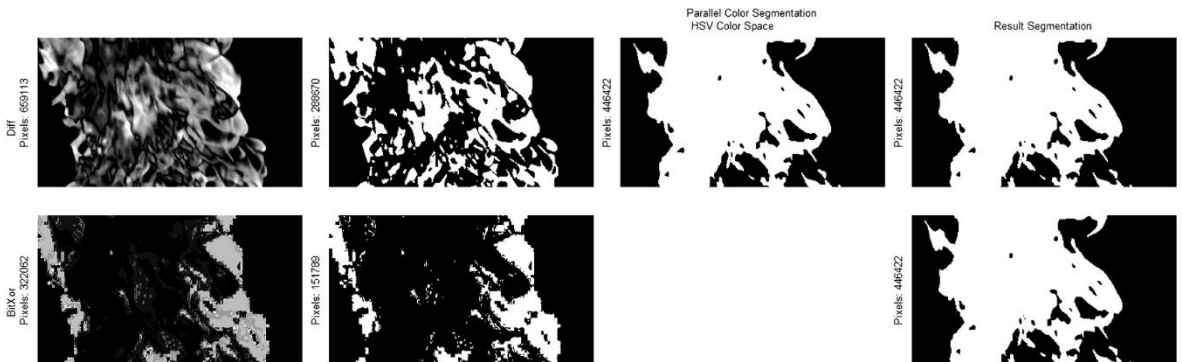
En la Tabla 13 se puede observar que el porcentaje de detección con el espacio de color HSV en la metodología en paralelo son más altos que en la metodología en cascada a pesar de no superar en la mayoría de los casos el 50% la segmentación. Con este espacio de color, también ambas topologías presentan problemas en la detección de imágenes de videos de flama azul.

La Figura 19 permite visualizar los resultados de la metodología en cascada vs topología en paralelo en el espacio de color HSV y las Figuras 20 y 21 observar la pantalla instantánea de resultados de las topologías en cascada y paralelo en el espacio de color HSV.

Figura 20. Topología cascada en espacio de color HSV



Figura 21. Topología paralelo en espacio de color HSV



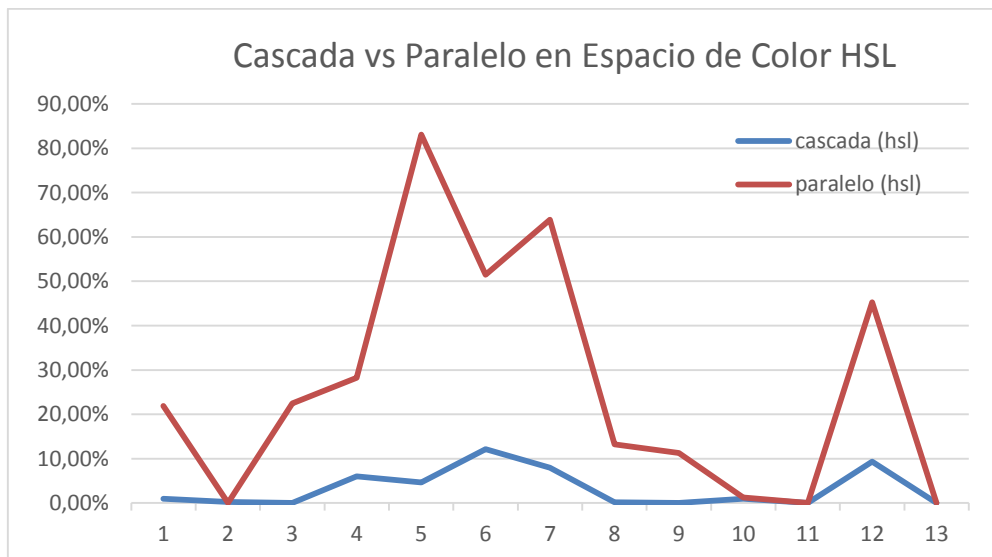
Para el espacio de color HSL, la Tabla 14 y Figura 22 presenta los resultados de la comparación de la topología cascada vs topología en paralelo.

Tabla 14. Resultados topología cascada vs topología en paralelo en espacio de color HSL

Video	Pixeles x Frame	cascada (%)	paralelo (%)
BG_Fire_Black_Screen_Effect	920986,57	1,00	21,91
Blue_Flames	362761,97	0,25	0,00
Blue_flame_loop	230399,79	0,00	22,47
Fractal_Noise_Fire_in_After_Effects_No_3rd_party_p	542729,32	6,03	28,22
Free_Stock_Footage	708012,97	4,67	83,08

Low_flames_flickering_with_a_black_background	70240,15	12,12	51,49
Realistic_CG_Fire	103231,09	7,96	63,92
Slow_Motion_Fire_Blaze_From_the_Bottom_Stock_Video	921598,21	0,20	13,23
Slow_Motion_Fire_HD_Motion_Background	921599,77	0,03	11,26
Slow_Motion_Fire_in_Blender	180068,89	0,95	1,23
Sony_Vegas_Blue_FIRE_Effect	6912,00	0,03	0,00
TGAVideo	27327,80	9,31	45,29
fire_blue	29329,03	0,04	0,00

Figura 22. Resultados topología cascada vs topología en paralelo en espacio de color HSL



En la Tabla 14 y figura 22, respectivamente, se puede observar que el porcentaje de detección con el espacio de color HSL en la metodología en paralelo son más altos que en la metodología en cascada mejorando la segmentación. Con este espacio de color, también ambas topologías presentan bajo porcentaje de detección de imágenes de videos de flama azul. Las Figuras 23 y 24 permiten observar la pantalla instantánea de resultados de las topologías en cascada y paralelo en el espacio de color HSL.

Figura 23. Topología cascada en espacio de color HSL

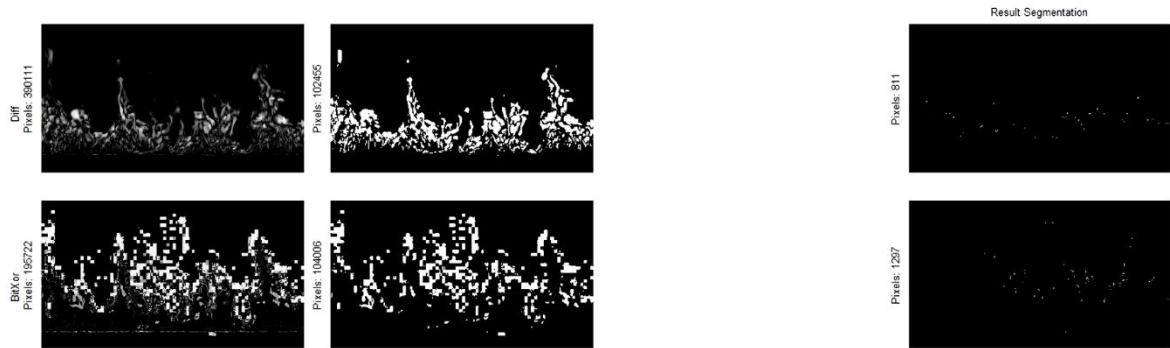
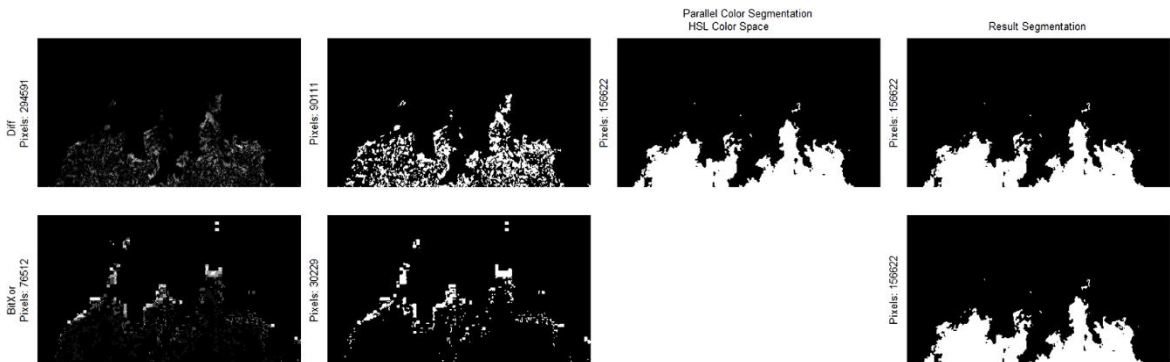


Figura 24. Topología paralelo en espacio de color HSL



En la Tabla 15 y Figura 25 se muestran los resultados de la segmentación en el espacio de color HSI y la comparación de la topología cascada vs topología en paralelo. En la Tabla 15 y Figura 25 se puede observar que el porcentaje de detección con el espacio de color HSI en la metodología en paralelo son más altos que en la metodología en cascada mejorando la segmentación que en otros espacios. Con este espacio de color, también ambas topologías presentan bajo porcentaje de detección de imágenes de videos de flama azul. La Figura 25 permite visualizar los resultados de la metodología en cascada vs topología en paralelo en el espacio de color HSL y las Figuras 26 y 27 observar la pantalla instantánea de resultados de las topologías en cascada y paralelo en el espacio de color HSL

Tabla 15. Resultados topología cascada vs topología en paralelo en espacio de color HSI

Video	Pixeles x Frame	cascada (%)	paralelo (%)
BG_Fire_Black_Screen_Effect	920986,57	1,04	20,98
Blue_Flames	362761,97	0,19	0,00
Blue_flame_loop	230399,79	0,00	59,91
Fractal_Noise_Fire_in_After_Effects_No_3rd_party_p	542729,32	7,02	40,78
Free_Stock_Footage	708012,97	4,83	82,69
Low_flames_flickering_with_a_black_background	70240,15	14,09	52,94
Realistic_CG_Fire	103231,09	9,57	64,26
Slow_Motion_Fire_Blaze_From_the_Bottom_Stock_Video	921598,21	0,22	13,27
Slow_Motion_Fire_HD_Motion_Background	921599,77	0,04	14,21
Slow_Motion_Fire_in_Blender	180068,89	0,87	15,61
Sony_Vegas_Blue_FIRE_Effect	6912,00	0,02	0,00
TGAVideo	27327,80	13,67	57,60
fire_blue	29329,03	0,00	0,00

Figura 25. Resultados topología cascada vs topología en paralelo en espacio de color HSI

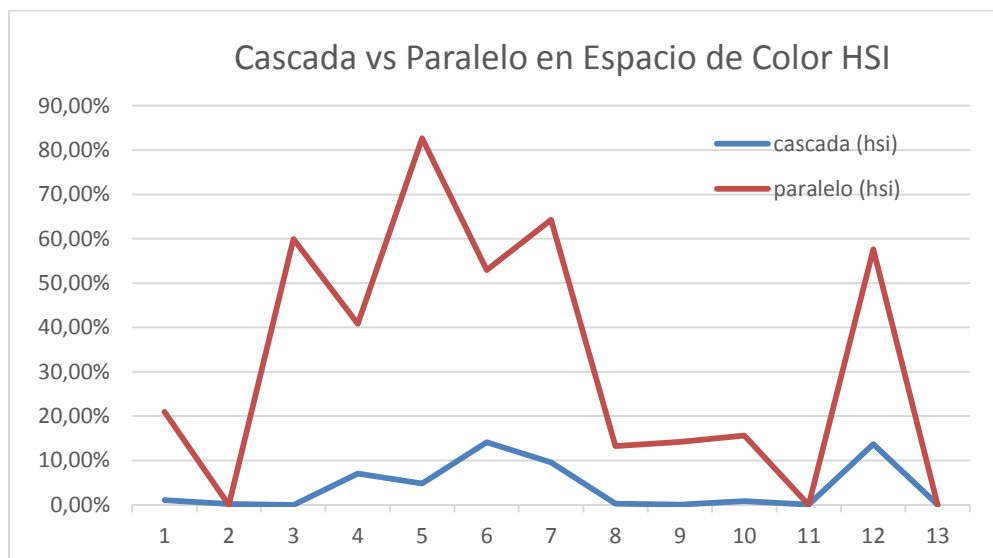


Figura 26. Topología cascada en espacio de color HSI

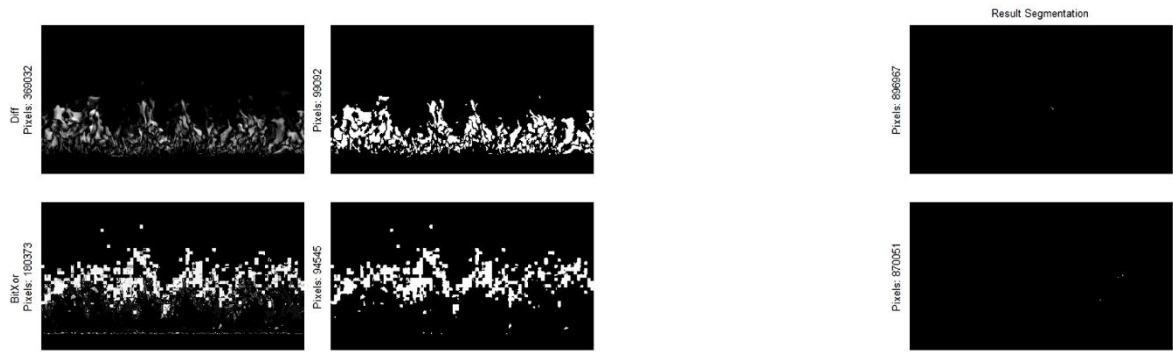
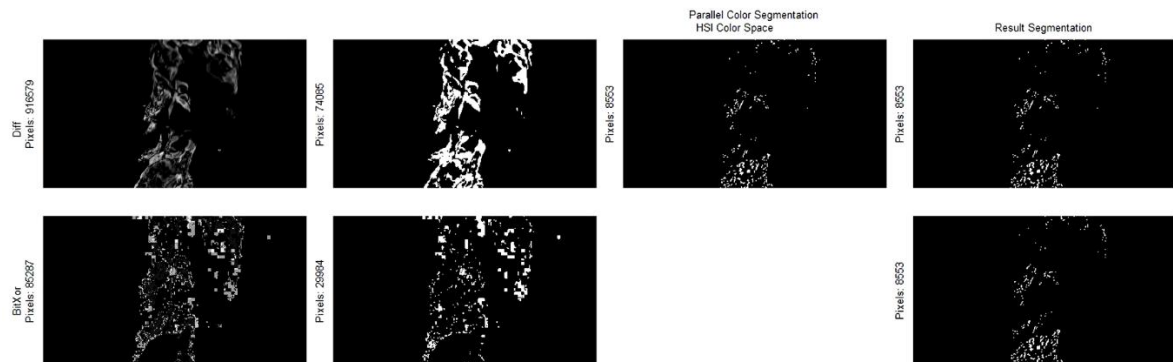


Figura 27. Topología paralelo en espacio de color HSI



La tabla 16, presenta la comparacion de los resultados de la metodología de paralelo menos (-) los resultados de la metodología en cascada, los valores negativos de la tabla se dan cuando la topología en cascada presenta un mejor resultado, posteriormente se realiza el cálculo del promedio de los resultados más óptimos en cascada y el promedio de los resultados más óptimos en paralelo.

Tabla 16. Resultados topología cascada vs topología en paralelo en los 7 espacios de color.

Video	Píxeles x Frame	Espacio de color (%), Paralelo – Cascada						
		rgb	ycbcr	cmy	cmyk	hsv	Hsl	hsi
BG_Fire_Black_Screen_Effect	920986,56	6,79352	6,764 57	24,41 054	- 0,001 39	28,40 429	20,91 286	19,94 254
Blue_Flames	362761,96	- 21,009 91	0,000 00	- 0,055 22	0,033 89	- 0,031 75	- 0,247 77	- 0,193 71
Blue_flame_loop	230399,78	50,593 57	0,002 35	- 0,003 08	0,000 37	- 0,002 80	22,46 644	59,91 144
Fractal_Noise_Fire_in_After_Effects_No_3rd_party_p	542729,31	5,9765 6	8,056 18	32,60 087	4,538 91	36,71 465	22,19 656	33,76 349
Free_Stock_Footage	708012,97	62,622 43	29,80 817	74,10 829	16,43 492	49,55 396	78,41 079	77,85 869
Low_flames_flickering_with_a_black_background	70240,15	- 52,214 39	5,721 89	30,30 578	0,699 44	64,32 998	39,37 168	38,84 274
Realistic_CG_Fire	103231,09	2,4177 0	1,752 50	41,98 450	3,662 92	57,36 268	55,96 728	54,69 602
Slow_Motion_Fire_Blaze_From_the_Bottom_Stock_Video	921598,21	9,8957 1	1,138 36	14,53 911	0,295 45	15,42 770	13,02 821	13,04 588
Slow_Motion_Fire_HD_Motion_Background	921599,76	13,751 17	0,723 52	16,14 832	0,070 89	17,14 734	11,23 073	14,16 525
Slow_Motion_Fire_in_Blender	180068,88	16,710 80	1,010 96	38,56 691	8,065 22	26,24 638	0,281 97	14,74 020
Sony_Vegas_Blue_FIRE_Effect	6912,00	- 43,443 02	- 0,004 37	- 0,064 50	0,012 96	- 0,449 40	- 0,028 94	- 0,017 33
TGAVideo	27327,80	0,0958 7	3,735 91	32,97 155	5,304 70	42,65 986	35,97 770	43,92 701
fire_blue	29329,02	- 30,246 66	0,000 00	- 0,147 05	0,110 77	- 0,219 18	- 0,035 49	- 0,003 67

En la Tabla 16, se puede observar que el promedio de los mejores resultados obtenidos en cascada es de 0.2198 y el promedio de los mejores resultados obtenidos en paralelo es de 3.55. Con estos datos se puede ver que la topología en paralelo es la metodología más asertiva para la detección de imagen de flama. Aquí se quiere resaltar que la mayor parte de los resultados más altos es de la topología en paralelo.

4.2 Extracción de características

La matriz características generada es de 2759 Señales de los 38 videos de la base de datos seleccionada x 84 características, las 2759 Señales corresponden al número de frame de los videos de la base de datos y las 84 características corresponden a 14 características sobre las imágenes originales, 14 características sobre las imágenes segmentadas y cada característica entrega 3 valores uno por cada capa de los espacios de color siendo un total de 84 valores. Se genera una matriz característica por topología, detección de movimiento y espacio de color. Para determinar cuáles de las características no presentaban variabilidad se realizó un análisis de relevancia al calcular la varianza, en este proceso de análisis de los resultados de la extracción de características se descartaron la media armónica y la media geométrica sobre las imágenes originales por que no presentaban variación en los resultados (todos los datos son cero o uno).

4.3 Clasificación.

4.3.1 Método de validación cruzada

El método de validación cruzada o “randomize bootstrap” es un método estadístico que se vale de un conjunto de datos, los cuales son divididos en dos subconjuntos, uno para el entrenamiento y otro para la validación del modelo de clasificación (Chernick, Murthy, & Nealy, 1985).

Otra variante del método conocida como validación cruzada múltiple (leave-one-out) consiste en generar un conjunto con N-1 muestras para el entrenamiento, dejando una muestra para la validación y repitiendo dicho proceso N veces (T. Wong, 2015).

Tabla 17. Descripción de parámetros en tablas de resultados.

Vn	Verdaderos negativos (el clasificador indica que no hay flama, en realidad no hay flama)
Vp	Verdaderos positivos (el clasificador indica que hay flama, en realidad hay flama)
Fp	Falsos positivos (el clasificador indica que hay flama pero en realidad no hay flama)
Fn	Falsos negativos (el clasificador indica que no hay flama, pero en realidad si hay flama)
Acc	Precisión del clasificador.

La Tabla 18 presenta los resultados de la clasificación en el espacio de color RGB basados en los 38 videos de la base de datos total.

Tabla 18. Resultados en RGB para la base de datos de 38 videos

	BitXor						Diff					
	Cascada			Paralelo			Cascada			Paralelo		
	knn	nb	svm	knn	nb	svm	knn	nb	svm	knn	nb	svm
Vp	1079	1079	1	1018	155	1068	1047	44	70	920	208	1074
Fp	1	1629	0	62	27	690	33	12	4	160	9	505
Fn	1629	1	1079	816	925	12	1674	1036	1010	679	872	6
Vn	50	49	1679	863	1645	989	5	1667	1675	1000	1661	1174
Acc(%)	40,92%	40,90%	60,89%	68,18%	65,41%	74,56%	38,13%	62,02%	63,25%	69,59%	67,96%	81,48%

En la Tabla 18 se puede observar que el porcentaje del índice de precisión del clasificador es más alto en las topologías en paralelo. El mayor índice de precisión es el de la topología en paralelo con la técnica de detección de movimiento Diff, bajo el clasificador SVM con 81.48%.

La Tabla 19 presenta los resultados del clasificador en el espacio de color YCbCr basados en los 38 videos de la base de datos.

Tabla 19. Resultados en YCBCR para la base de datos de 38 videos

	BitXor						Diff					
	Cascada			Paralelo			Cascada			Paralelo		
	knn	nb	svm	knn	nb	svm	knn	nb	svm	knn	nb	svm
Vp	645	809	966	989	52	1067	753	176	1024	1021	93	1075
Fp	435	434	456	91	31	562	327	36	535	59	36	555
Fn	1044	264	114	654	1028	13	1006	904	56	628	987	5
Vn	635	1203	1223	1025	1648	1117	673	1633	1144	1051	1643	1124
Acc(%)	46,39	74,24	79,34	73,00	61,62	79,16	51,69	65,81	78,58	75,10	62,92	79,70
)	%	%	%	%	%	%	%	%	%	%	%	%

En la Tabla 19 se observa el porcentaje del índice de precisión del clasificador es más alto en las topologías en paralelo siendo este muy ajustado en algunos casos con la topología en cascada. De estos resultados se puede concluir que el mayor índice de precisión es el de la topología en paralelo con la técnica de detección de movimiento Diff, bajo el clasificador SVM con 79.70%.

La Tabla 20 presenta los resultados del clasificador en el espacio de color CMY basados en los 38 videos de la base de datos.

Tabla 20. Resultados en CMY para la base de datos de 38 videos

	BitXor						Diff					
	Cascada			Paralelo			Cascada			Paralelo		
	knn	nb	svm	knn	nb	svm	knn	nb	svm	knn	nb	svm
Vp	1080	0	0	1023	436	492	1080	0	0	1016	394	1073
Fp	0	0	0	57	55	2	0	0	0	64	39	550
Fn	1679	1080	1080	722	644	588	1679	1080	1080	688	686	7
Vn	0	1679	1679	957	1602	1677	0	1679	1679	991	1622	1129
Acc(%)	39,14	60,86	60,86	71,77	74,46	78,62	39,14	60,86	60,86	72,74	73,55	79,81
)	%	%	%	%	%	%	%	%	%	%	%	%

En la Tabla 20 se puede observar que el porcentaje del índice de precisión del clasificador es más alto en las topologías en paralelo. El mayor índice de precisión es el de la topología en paralelo con la técnica de detección de movimiento Diff, bajo el clasificador SVM con 79.81%.

La Tabla 21 presenta los resultados del clasificador en el espacio de color CMYK basados en los 38 videos de la base de datos.

Tabla 21. Resultados en CMYK para la base de datos de 38 videos

	BitXor						Diff					
	Cascada			Paralelo			Cascada			Paralelo		
	knn	nb	svm	knn	nb	svm	knn	nb	svm	knn	nb	svm
Vp	1080	0	0	1051	907	908	1080	0	0	1056	844	844
Fp	0	0	0	29	57	3	0	0	0	24	25	1
Fn	1679	1080	1080	487	173	172	1679	1080	1080	456	236	236
Vn	0	1679	1679	1192	1618	1676	0	1679	1679	1223	1647	1678
Acc(%)	39,14%	60,86%	60,86%	81,30%	91,65%	93,66%	39,14%	60,86%	60,86%	82,60%	90,52%	91,41%

De la Tabla 21 se puede observar que el porcentaje del índice de precisión del clasificador es más alto en las topologías en paralelo en ambas topologías. Sin embargo, el mayor índice de precisión es el de la topología en paralelo con la técnica de detección de movimiento Bitxor, bajo el clasificador SVM con 93.66%.

La Tabla 22 presenta los resultados del clasificador en el espacio de color HSV basados en los 38 videos de la base de datos.

Tabla 22. Resultados en HSV para la base de datos de 38 videos

	BitXor						Diff					
	Cascada			Paralelo			Cascada			Paralelo		
	knn	nb	svm	knn	nb	svm	knn	nb	svm	knn	nb	svm
Vp	736	344	332	1046	119	1063	527	958	627	1047	432	1063
Fp	344	58	3	34	9	477	553	699	13	33	18	450
Fn	1239	736	748	538	961	17	1023	113	453	494	648	17
Vn	440	1614	1676	1141	1670	1202	656	964	1666	1185	1661	1229
Acc(%)	42,62	71,15	72,78	79,27	64,84	82,09	42,88	70,30	83,11	80,90	75,86	83,07
)	%	%	%	%	%	%	%	%	%	%	%	%

En la Tabla 22 se puede observar que el porcentaje del índice de precisión del clasificador es más alto en las topologías en paralelo. El mayor índice de precisión es el de la topología en cascada con la técnica de detección de movimiento Diff, bajo el clasificador SVM con 83.11%.

La Tabla 23 presenta los resultados del clasificador en el espacio de color HSL basados en los 38 videos de la base de datos.

Tabla 23. Resultados en HSL para la base de datos de 38 videos.

	BitXor						Diff					
	Cascada			Paralelo			Cascada			Paralelo		
	knn	nb	svm	knn	nb	svm	knn	nb	svm	knn	nb	svm
Vp	722	356	346	1041	1016	1072	484	965	656	1012	1044	805
Fp	358	64	7	39	444	439	596	721	12	68	428	0
Fn	1226	724	734	503	64	8	1136	109	424	484	34	275
Vn	453	1600	1672	1176	1232	1240	543	941	1667	1195	1249	1679
Acc(%)	42,59	71,28	73,14	80,36	81,57	83,80	37,22	69,66	84,20	79,99	83,23	90,03
)	%	%	%	%	%	%	%	%	%	%	%	%

En la Tabla 23 se puede observar que el porcentaje del índice de presicion del clasificador es más alto en las topologías en paralelo. El mayor índice de presicion es el de la topología en paralelo con la técnica de detección de movimiento Diff, bajo el clasificador SVM con 90.03%.

La Tabla 24 presenta los resultados del clasificador en el espacio de color HSI basados en los 38 videos de la base de datos.

Tabla 24. Resultados en HSI para la base de datos de 38 videos

	BitXor						Diff					
	Cascada			Paralelo			Cascada			Paralelo		
	knn	nb	svm	knn	nb	svm	knn	nb	svm	knn	nb	svm
Vp	101	1049	293	917	870	872	206	447	911	943	675	688
Fp	979	679	81	163	162	3	874	37	296	137	79	0
Fn	1173	31	787	774	208	208	1149	616	169	701	401	392
Vn	506	980	1598	905	1503	1676	530	1615	1383	978	1574	1679
Acc(%)	22,00	74,08	68,54	66,04	86,51	92,35	26,68	75,95	83,15	69,63	82,41	85,79
)	%	%	%	%	%	%	%	%	%	%	%	%

En la Tabla 23 se puede observar que el porcentaje del índice de presicion del clasificador es más alto en las topologías en paralelo. El mayor índice de presicion es el de la topología

en paralelo con la técnica de detección de movimiento Bitxor, bajo el clasificador SVM con 92.35%.

Del conjunto de resultados de la validación cruzada anteriormente presentados se observa que el mayor índice de precisión es el de la topología en paralelo con la técnica de detección de movimiento Bitxor, bajo el clasificador SVM con 91.41%.

En busca de mejorar el conjunto de resultados de la validación anterior, se realizan pruebas para tratar de mejorar el resultado en la precisión del clasificador. Por lo tanto, se realiza el análisis de los 38 videos, pero en este caso separando las matrices, en el resultado anteriormente presentado se trabajó con las matrices de los videos originales y los videos segmentados unidas, sin embargo, a continuación, se presentará el análisis de las matrices por separado.

La Tabla 23 presenta los resultados del clasificador en el espacio de color HSI basados en los 38 videos de la base de datos. En la Tabla 23 se puede observar que el porcentaje del índice de precisión del clasificador es más alto en las topologías en paralelo. El mayor índice de precisión es el de la topología en paralelo con la técnica de detección de movimiento Bitxor, bajo el clasificador SVM con 92.35%.

La Tabla 25 presenta los resultados del clasificador en el espacio de color RGB basados en los 38 videos de la base de datos en la matriz de los videos originales. En la Tabla 24 se puede observar que el porcentaje del índice de precisión del clasificador es similar al presentado anteriormente donde se trabajó con las matrices del video original y el video segmentado unidas. El mayor índice de precisión es el de la topología en paralelo con la técnica de detección de movimiento Diff, bajo el clasificador SVM con 81.48%, el mismo que el presentado en la tabla 17.

Tabla 25. Resultados en RGB para la base de datos de 38 videos en la matriz de videos originales

	BitXor						Diff					
	Cascada			Paralelo			Cascada			Paralelo		
	knn	nb	svm	knn	nb	svm	knn	nb	svm	knn	nb	svm
Vp	1079	0	1	1034	112	1070	1040	1077	70	938	183	1074

Fp	1	0	0	46	10	691	40	1666	3	142	28	505
Fn	1626	1080	1079	775	968	10	1679	3	1010	621	897	6
Vn	53	1679	1679	904	1664	988	0	13	1676	1058	1649	1174
Acc(%))	41,03 %	60,86 %	60,89 %	70,24 %	64,49 %	74,59 %	37,69 %	39,51 %	63,28 %	72,35 %	66,45 %	81,48 %

La Tabla 26 presenta los resultados del clasificador en el espacio de color RGB basados en los 38 videos de la base de datos en la matriz de los videos Segmentados. En la Tabla 25 se puede observar que el porcentaje del índice de presicion del clasificador varia solo un poco al presentado anteriormente donde se trabajo con las matrices del video original y el video segentado unidas. El mayor índice de presicion es el de la topología en paralelo con la técnica de detección de movimiento Diff, bajo el clasificador SVM con 80.57% menor al del video original en un 0.91% pero no es significativa la diferencia

Tabla 26. Resultados en RGB para la base de datos de 38 videos en la matriz de videos segmentados

	BitXor						Diff					
	Cascada			Paralelo			Cascada			Paralelo		
	knn	nb	svm	knn	nb	svm	knn	nb	svm	knn	nb	svm
Vp	1079	1079	1	1001	170	1046	1044	23	70	913	175	1080
Fp	1	1635	0	79	51	697	36	9	5	167	14	536
Fn	1628	1	1079	851	910	34	1673	1057	1010	687	905	0
Vn	51	44	1679	828	1627	982	6	1670	1674	992	1664	1143
Acc(%))	40,96 %	40,70 %	60,89 %	66,29 %	65,16 %	73,50 %	38,06 %	61,36 %	63,21 %	69,05 %	66,68 %	80,57 %

La Tabla 27 presenta los resultados del clasificador en el espacio de color YCbCr basados en los 38 videos de la base de datos en la matriz de los videos originales. En la Tabla 26 se puede observar que el porcentaje del índice de presicion del clasificador es un poco menor al presentado anteriormente donde se trabajo con las matrices del video original y el video segmentado unidas. El mayor índice de presicion es el de la topología en paralelo con la técnica de detección de movimiento Diff, bajo el clasificador SVM con 79.67%.

Tabla 27. Resultados en YCBCR para la base de datos de 38 videos en la matriz de videos originales

BitXor	Diff
--------	------

	Cascada			Paralelo			Cascada			Paralelo		
	knn	nb	svm	knn	nb	svm	knn	nb	svm	knn	nb	svm
Vp	579	51	873	989	159	1068	665	34	1003	1036	47	1074
Fp	501	19	476	91	17	561	415	70	563	44	19	555
Fn	1197	1029	207	656	921	12	1109	1046	77	620	1033	6
Vn	482	1623	1203	1023	1662	1118	570	1601	1116	1059	1660	1124
Acc(%))	38,46 %	61,50 %	75,24 %	72,92 %	66,00 %	79,23 %	44,76 %	59,43 %	76,80 %	75,93 %	61,87 %	79,67 %

La Tabla 28 presenta los resultados del clasificador en el espacio de color YCbCr basados en los 38 videos de la base de datos en la matriz de los videos Segmentados. En la Tabla 27 se puede observar que el porcentaje del índice de presicion del clasificador varia solo un poco al presentado anteriormente donde se trabajo con las matrices del video original y el video segentado unidas. El mayor índice de presicion es el de la topología en paralelo con la técnica de detección de movimiento Diff, bajo el clasificador SVM con 77.82% menor al del video original en un 2.75% pero no es significativa la diferencia.

Tabla 28. Resultados en YBCRCR para la base de datos de 38 videos en la matriz de videos segmentados

	BitXor						Diff					
	Cascada			Paralelo			Cascada			Paralelo		
	knn	nb	svm	knn	nb	svm	knn	nb	svm	knn	nb	svm
Vp	642	1031	710	962	52	1033	746	210	748	1012	97	1044
Fp	438	450	477	118	37	562	334	56	543	68	36	576
Fn	1051	47	370	679	1028	47	1001	870	332	646	983	36
Vn	628	1205	1202	1000	1642	1117	678	1618	1136	1033	1643	1103
Acc(%))	46,03 %	81,81 %	69,30 %	71,11 %	61,40 %	77,93 %	51,61 %	66,38 %	68,29 %	74,12 %	63,07 %	77,82 %

La Tabla 29 presenta los resultados del clasificador en el espacio de color CMY basados se puede observar que el porcentaje del índice de presicion del clasificador es un poco menor al presentado anteriormente donde se trabajo con las matrices del video original y el video segentado unidas. El mayor índice de presicion es el de la topología en paralelo con la técnica de detección de movimiento Diff, bajo el clasificador SVM con 79.96%.

Tabla 29. Resultados en CMY para la base de datos de 38 videos en la matriz de videos originales

	BitXor						Diff					
	Cascada			Paralelo			Cascada			Paralelo		
	knn	nb	svm	knn	nb	svm	knn	nb	svm	knn	nb	svm
Vp	1080	0	0	986	458	496	1080	0	0	986	399	1078
Fp	0	0	0	94	66	2	0	0	0	94	46	551
Fn	1679	1080	1080	759	622	584	1679	1080	1080	686	681	2
Vn	0	1679	1679	920	1613	1677	0	1679	1679	993	1633	1128
Acc(%))	39,14 %	60,86 %	60,86 %	69,08 %	75,06 %	78,76 %	39,14 %	60,86 %	60,86 %	71,73 %	73,65 %	79,96 %

La Tabla 30 presenta los resultados del clasificador en el espacio de color CMY basados en los 38 videos de la base de datos en la matriz de los videos Segmentados. En la Tabla 29 se puede observar que el porcentaje del índice de presicion del clasificador varia solo un poco al presentado anteriormente donde se trabajo con las matrices del video original y el video segentado unidas. El mayor índice de presicion es el de la topología en paralelo con la técnica de detección de movimiento Diff, bajo el clasificador SVM con 79.63% menor al del video original en un 0.33% pero no es significativa la diferencia.

Tabla 30. Resultados en CMY para la base de datos de 38 videos en la matriz de videos segmentados

	BitXor						Diff					
	Cascade			Parallel			Cascade			Parallel		
	knn	nb	svm	knn	nb	svm	knn	nb	svm	knn	nb	svm
Vp	1080	0	0	964	398	489	1080	0	0	947	355	1075
Fp	0	0	0	116	177	3	0	0	0	133	124	557
Fn	1679	1080	1080	775	682	591	1679	1080	1080	739	725	5
Vn	0	1679	1679	904	1490	1676	0	1679	1679	940	1541	1122
Acc(%))	39,14 %	60,86 %	60,86 %	67,71 %	68,73 %	78,47 %	39,14 %	60,86 %	60,86 %	68,39 %	69,07 %	79,63 %

La Tabla 31 presenta los resultados del clasificador en el espacio de color CMYK basados en los 38 videos de la base de datos en la matriz de los videos originales. En la Tabla 30 se puede observar que el porcentaje del índice de presicion del clasificador es un poco mayor al presentado anteriormente donde se trabajo con las matrices del video original y el video segentado unidas El mayor índice de presicion es el de la topología en

La Tabla 33 presenta los resultados del clasificador en el espacio de color HSV basados en los 38 videos de la base de datos en la matriz de los videos originales. En la Tabla 32 se puede observar que el porcentaje del índice de precision del clasificador es un poco menor al presentado anteriormente donde se trabajo con las matrices del video original y el video segentado unidas. El mayor índice de precision es el de la topología en paralelo con la técnica de detección de movimiento Diff, bajo el clasificador SVM con 83.25% Cuando en la tabla 21 donde se trabajo con las matrices del video original y el video segentado unidas en el mismo espacio de color el mayor índice de precision es el de la topología en cascada con la técnica de detección de movimiento Diff, bajo el clasificador SVM con 83.11%.

Tabla 33. Resultados en HSV para la base de datos de 38 videos en la matriz de videos originales

	BitXor						Diff					
	Cascade			Parallel			Cascade			Parallel		
	knn	nb	svm	knn	nb	svm	knn	nb	svm	knn	nb	svm
Vp	719	275	302	991	41	1066	492	917	607	1029	1034	1070
Fp	361	351	3	89	5	476	588	699	33	51	456	452
Fn	1329	805	778	548	1039	14	1153	158	473	500	46	10
Vn	350	1320	1676	1131	1674	1203	526	962	1646	1179	1223	1227
Acc(%)	38,75%	57,98%	71,69%	76,91%	62,16%	82,24%	36,90%	68,68%	81,66%	80,03%	81,81%	83,25%

La Tabla 34 presenta los resultados del clasificador en el espacio de color HSV basados en los 38 videos de la base de datos en la matriz de los videos Segmentados. En la Tabla 33 se puede observar que el porcentaje del índice de precision del clasificador varia solo un poco al presentado anteriormente donde se trabajo con las matrices del video original y el video segentado unidas El mayor índice de precision es el de la topología en paralelo con la técnica de detección de movimiento Diff, bajo el clasificador SVM con 82.53% menor al del video original en un 0.72% pero no es significativa la diferencia.

Tabla 34. Resultados en HSV para la base de datos de 38 videos en la matriz de videos segmentados

	BitXor						Diff					
	Cascade			Parallel			Cascade			Parallel		
	knn	nb	svm	knn	nb	svm	knn	nb	svm	knn	nb	svm

Vp	740	352	160	1022	120	1054	530	929	409	1020	337	1059
Fp	340	72	64	58	19	482	550	729	93	60	15	461
Fn	1239	728	920	581	960	26	1052	146	671	532	743	21
Vn	440	1603	1615	1098	1660	1197	627	949	1586	1147	1664	1218
Acc(%)	42,77	70,96	64,33	76,84	64,52	81,59	41,94	68,22	72,31	78,54	72,53	82,53
)	%	%	%	%	%	%	%	%	%	%	%	%

La Tabla 35 presenta los resultados del clasificador en el espacio de color HSL basados en los 38 videos de la base de datos en la matriz de los videos originales. En la Tabla 34 se puede observar que el porcentaje del índice de precisión del clasificador es un poco menor al presentado anteriormente donde se trabajo con las matrices del video original y el video segmentado unidas. De estos resultados podemos concluir que el mayor índice de precisión es el de la topología en paralelo con la técnica de detección de movimiento Diff, bajo el clasificador SVM con 89.89%.

Tabla 35. Resultados en HSL para la base de datos de 38 videos en la matriz de videos originales

	BitXor						Diff					
	Cascade			Parallel			Cascade			Parallel		
	knn	nb	svm	knn	nb	svm	knn	nb	svm	knn	nb	svm
Vp	700	201	330	1020	1027	1073	474	925	633	1015	1053	801
Fp	380	32	10	60	453	439	606	717	38	65	474	0
Fn	1322	879	750	504	53	7	1087	151	447	501	26	279
Vn	357	1635	1669	1175	1225	1240	592	947	1641	1178	1204	1679
Acc(%)	38,31	66,84	72,45	79,56	81,65	83,83	38,64	68,32	82,42	79,49	81,86	89,89
)	%	%	%	%	%	%	%	%	%	%	%	%

La Tabla 36 presenta los resultados del clasificador en el espacio de color HSV basados en los 38 videos de la base de datos en la matriz de los videos Segmentados. En la Tabla 35 se puede observar que el porcentaje del índice de precisión del clasificador varia solo un poco al presentado anteriormente donde se trabajo con las matrices del video original y el video segmentado unidas. El mayor índice de precisión es el de la topología en paralelo con la técnica de detección de movimiento Diff, bajo el clasificador SVM con 87.50% menor al del video original en un 2.39% pero no es significativa la diferencia.

Tabla 36. Resultados en HSL para la base de datos de 38 videos en la matriz de videos segmentados

	BitXor						Diff					
	Cascade			Parallel			Cascade			Parallel		
	knn	nb	svm	knn	nb	svm	knn	nb	svm	knn	nb	svm
Vp	728	356	206	1032	928	1063	479	937	454	997	859	795
Fp	352	65	90	48	452	487	601	794	133	83	443	60
Fn	1227	724	874	539	151	17	1131	142	626	504	221	285
Vn	452	1609	1589	1140	1226	1192	548	883	1546	1175	1234	1619
Acc(%)	42,77	71,35	65,06	78,72	78,13	81,73	37,22	66,04	72,49	78,72	75,92	87,50
)	%	%	%	%	%	%	%	%	%	%	%	%

La Tabla 37 presenta los resultados del clasificador en el espacio de color HSI basados en los 38 videos de la base de datos en la matriz de los videos originales. En la Tabla 36 se puede observar que el porcentaje del índice de precision del clasificador es un poco menor al presentado anteriormente donde se trabajo con las matrices del video original y el video segentado unidas. El mayor índice de precision es el de la topología en paralelo con la técnica de detección de movimiento Bitxor, bajo el clasificador SVM con 92.13%.

Tabla 37. Resultados en HSI para la base de datos de 38 videos en la matriz de videos originales

	BitXor						Diff					
	Cascade			Parallel			Cascade			Parallel		
	knn	nb	svm	knn	nb	svm	knn	nb	svm	knn	nb	svm
Vp	896	1076	162	977	808	871	565	350	513	987	560	687
Fp	184	1582	18	103	282	8	515	45	53	93	110	6
Fn	1290	4	918	739	272	209	1244	712	567	648	520	393
Vn	389	97	1661	940	1396	1671	435	1612	1626	1031	1569	1673
Acc(%)	46,57	42,52	66,07	69,48	79,91	92,13	36,25	72,16	77,53	73,14	77,17	85,54
)	%	%	%	%	%	%	%	%	%	%	%	%

La siguiente tabla 38 presenta los resultados del clasificador en el espacio de color HSV basados en los 38 videos de la base de datos en la matriz de los videos Segmentados. En la Tabla 38 se puede observar que el porcentaje del índice de precision del clasificador varia solo un poco al presentado anteriormente donde se trabajo con las matrices del video original y el video segentado unidas. El mayor índice de precision es el de la topología en paralelo con la técnica de detección de movimiento Diff, bajo el clasificador SVM con 90.21% menor al del video original en un 1.92% pero no es significativa la diferencia.

Tabla 38. Resultados en HSI para la base de datos de 38 videos en la matriz de videos segmentados

	BitXor						Diff					
	Cascade			Parallel			Cascade			Parallel		
	knn	nb	svm	knn	nb	svm	knn	nb	svm	knn	nb	svm
Vp	111	1057	239	839	833	849	205	848	815	881	679	670
Fp	969	714	105	241	112	39	875	644	412	199	82	21
Fn	1202	23	841	858	241	231	1160	227	265	789	398	410
Vn	477	948	1574	821	1559	1640	519	1027	1267	890	1568	1658
Acc(%)	21,31	73,12	65,71	60,17	87,14	90,21	26,24	68,28	75,46	64,19	82,40	84,38
)	%	%	%	%	%	%	%	%	%	%	%	%

Posteriormente, con el fin de evitar falsos negativos, se eliminan los videos de la base de datos de fuego azul, y se conforma la base de datos de videos de flama amarilla de fondo negro, videos de fuego de las bases de datos y videos que no tienen fuego, como se puede observar en las tablas del 39 al 59 no hay mejoría en los resultados obtenidos.

Tabla 39. Resultados en RGB para la base de datos de videos exceptuando los videos de flama azul en la matriz de videos originales.

	BitXor						Diff					
	Cascade			Parallel			Cascade			Parallel		
	knn	nb	svm	knn	nb	svm	knn	nb	svm	knn	nb	svm
Vp	1079	1079	1	1002	65	1045	1044	23	70	910	149	1080
Fp	1	1439	0	78	11	503	36	9	6	170	10	342
Fn	1433	1	1079	646	1015	35	1478	1057	1010	494	931	0
Vn	51	45	1484	838	1473	981	6	1475	1478	990	1473	1142
Acc(%)	44,07	43,84	57,92	71,76	59,98	79,02	40,95	58,42	60,37	74,10	63,29	86,66
)	%	%	%	%	%	%	%	%	%	%	%	%

Tabla 40. Resultados en RGB para la base de datos de videos exceptuando los videos de flama azul en la matriz de videos segmentados.

	BitXor						Diff					
	Cascade			Parallel			Cascade			Parallel		
	knn	nb	svm	knn	nb	svm	knn	nb	svm	knn	nb	svm
Vp	1079	0	1	1031	82	1072	1037	51	70	952	219	1077
Fp	1	1	0	49	3	498	43	27	5	128	9	312
Fn	1431	1080	1079	579	998	8	1484	1029	1010	418	861	3
Vn	53	1483	1484	905	1478	986	0	1457	1479	1066	1474	1172

Tabla 44. Resultados en YCBCR para la base de datos de videos exceptuando los videos de flama azul en la matriz de videos originales y segmentados unidos.

	BitXor						Diff					
	Cascade			Parallel			Cascade			Parallel		
	knn	nb	svm	knn	nb	svm	knn	nb	svm	knn	nb	svm
Vp	667	894	980	993	40	1075	770	182	1041	1021	87	1074
Fp	413	401	426	87	20	367	310	28	500	59	17	360
Fn	942	183	100	465	1040	5	874	898	39	427	993	6
Vn	542	1042	1058	1019	1464	1117	610	1451	984	1057	1467	1124
Acc(%)	47,15	76,83	79,49	78,47	58,66	85,49	53,82	63,81	78,98	81,05	60,61	85,73
)	%	%	%	%	%	%	%	%	%	%	%	%

Tabla 45. Resultados en CMY para la base de datos de videos exceptuando los videos de flama azul en la matriz de videos originales.

	BitXor						Diff					
	Cascade			Parallel			Cascade			Parallel		
	knn	nb	svm	knn	nb	svm	knn	nb	svm	knn	nb	svm
Vp	1080	0	0	966	390	1069	1080	0	0	952	353	1073
Fp	0	0	0	114	168	533	0	0	0	128	68	472
Fn	1484	1080	1080	696	690	11	1484	1080	1080	664	727	7
Vn	0	1484	1484	788	1308	951	0	1484	1484	820	1406	1012
Acc(%)	42,12	57,88	57,88	68,41	66,43	78,78	42,12	57,88	57,88	69,11	68,87	81,32
)	%	%	%	%	%	%	%	%	%	%	%	%

Tabla 46. Resultados en CMY para la base de datos de videos exceptuando los videos de flama azul en la matriz de videos segmentados.

	BitXor						Diff					
	Cascade			Parallel			Cascade			Parallel		
	knn	nb	svm	knn	nb	svm	knn	nb	svm	knn	nb	svm
Vp	1080	0	0	996	378	1074	1080	0	0	1000	321	1079
Fp	0	0	0	84	30	532	0	0	0	80	11	472
Fn	1484	1080	1080	677	702	6	1484	1080	1080	606	759	1
Vn	0	1484	1484	807	1454	952	0	1484	1484	878	1473	1012
Acc(%)	42,12	57,88	57,88	70,32	71,45	79,02	42,12	57,88	57,88	73,24	69,97	81,55
)	%	%	%	%	%	%	%	%	%	%	%	%

Tabla 47. Resultados en CMY para la base de datos de videos exceptuando los videos de flama azul en la matriz de videos originales y segmentados unidos.

	BitXor						Diff					
	Cascade			Parallel			Cascade			Parallel		
	knn	nb	svm	knn	nb	svm	knn	nb	svm	knn	nb	svm
Vp	1080	0	0	1029	387	1075	1080	0	0	1016	364	1076
Fp	0	0	0	51	24	529	0	0	0	64	12	470
Fn	1484	1080	1080	652	693	5	1484	1080	1080	606	716	4
Vn	0	1484	1484	832	1452	955	0	1484	1484	878	1467	1014
Acc(%)	42,12	57,88	57,88	72,58	71,95	79,17	42,12	57,88	57,88	73,87	71,55	81,51
)	%	%	%	%	%	%	%	%	%	%	%	%

Tabla 48. Resultados en CMYK para la base de datos de videos exceptuando los videos de flama azul en la matriz de videos originales.

	BitXor						Diff					
	Cascade			Parallel			Cascade			Parallel		
	knn	nb	svm	knn	nb	svm	knn	nb	svm	knn	nb	svm
Vp	1080	0	0	1039	797	898	1080	0	0	1048	759	1070
Fp	0	0	0	41	129	0	0	0	0	32	106	224
Fn	1484	1080	1080	309	283	182	1484	1080	1080	280	321	10
Vn	0	1484	1484	1175	1348	1484	0	1484	1484	1204	1376	1260
Acc(%)	42,12	57,88	57,88	86,35	83,89	92,90	42,12	57,88	57,88	87,83	83,33	90,87
)	%	%	%	%	%	%	%	%	%	%	%	%

Tabla 49. Resultados en CMYK para la base de datos de videos exceptuando los videos de flama azul en la matriz de videos segmentados.

	BitXor						Diff					
	Cascade			Parallel			Cascade			Parallel		
	knn	nb	svm	knn	nb	svm	knn	nb	svm	knn	nb	svm
Vp	1080	0	0	1043	865	912	1080	0	0	1052	1047	1077
Fp	0	0	0	37	42	5	0	0	0	28	297	230
Fn	1484	1080	1080	304	215	168	1484	1080	1080	296	33	3
Vn	0	1484	1484	1180	1441	1479	0	1484	1484	1188	1187	1254
Acc(%)	42,12	57,88	57,88	86,70	89,97	93,25	42,12	57,88	57,88	87,36	87,13	90,91
)	%	%	%	%	%	%	%	%	%	%	%	%

Tabla 50. Resultados en CMYK para la base de datos de videos exceptuando los videos de flama azul en la matriz de videos originales y segmentados unidos.

	BitXor						Diff					
	Cascade			Parallel			Cascade			Parallel		

	knn	nb	svm	knn	nb	svm	knn	nb	svm	knn	nb	svm
Vp	1080	0	0	1050	900	903	1080	0	0	1055	1059	1072
Fp	0	0	0	30	22	0	0	0	0	25	248	224
Fn	1484	1080	1080	294	180	177	1484	1080	1080	258	21	8
Vn	0	1484	1484	1190	1459	1484	0	1484	1484	1226	1235	1260
Acc(%))	42,12 %	57,88 %	57,88 %	87,36 %	92,11 %	93,10 %	42,12 %	57,88 %	57,88 %	88,96 %	89,50 %	90,95 %

Tabla 51. Resultados en HSV para la base de datos de videos exceptuando los videos de flama azul en la matriz de videos originales.

	BitXor						Diff					
	Cascade			Parallel			Cascade			Parallel		
	knn	nb	svm	knn	nb	svm	knn	nb	svm	knn	nb	svm
Vp	762	367	230	1016	106	1059	541	946	443	1018	340	1060
Fp	318	393	92	64	2	342	539	633	112	62	7	320
Fn	1099	713	850	442	974	21	970	131	637	404	740	20
Vn	385	1087	1392	1042	1482	1142	514	847	1372	1080	1477	1164
Acc(%))	44,73 %	56,80 %	63,26 %	80,27 %	61,93 %	85,84 %	41,15 %	70,12 %	70,79 %	81,83 %	70,87 %	86,74 %

Tabla 52. Resultados en HSV para la base de datos de videos exceptuando los videos de flama azul en la matriz de videos segmentados.

	BitXor						Diff					
	Cascade			Parallel			Cascade			Parallel		
	knn	nb	svm	knn	nb	svm	knn	nb	svm	knn	nb	svm
Vp	736	1079	310	1000	48	1067	504	899	611	1035	1022	1074
Fp	344	1265	3	80	7	337	576	606	29	45	315	314
Fn	1194	1	770	409	1032	13	1036	173	469	362	58	6
Vn	290	219	1481	1075	1477	1147	448	871	1455	1122	1168	1170
Acc(%))	40,02 %	50,62 %	69,85 %	80,93 %	59,48 %	86,35 %	37,13 %	69,44 %	80,58 %	84,13 %	85,45 %	87,52 %

Tabla 53. Resultados en HSV para la base de datos de videos exceptuando los videos de flama azul en la matriz de videos originales y segmentados unidos.

	BitXor						Diff					
	Cascade			Parallel			Cascade			Parallel		
	knn	nb	svm	knn	nb	svm	knn	nb	svm	knn	nb	svm
Vp	759	1079	349	1043	134	1065	554	943	1023	1046	433	1068
Fp	321	1272	3	37	3	338	526	600	431	34	12	312

Fn	1103	1	731	394	946	15	954	124	57	358	647	12
Vn	381	212	1481	1090	1481	1146	530	872	1053	1126	1472	1172
Acc(%)	44,46	50,35	71,37	83,19	62,99	86,23	42,28	71,48	80,97	84,71	74,30	87,36
)	%	%	%	%	%	%	%	%	%	%	%	%

Tabla 54. Resultados en HSI para la base de datos de videos exceptuando los videos de flama azul en la matriz de videos originales.

	BitXor						Diff					
	Cascade			Parallel			Cascade			Parallel		
	knn	nb	svm	knn	nb	svm	knn	nb	svm	knn	nb	svm
Vp	740	398	223	1032	894	1065	489	954	466	994	825	1067
Fp	340	515	85	48	312	352	591	756	123	86	307	321
Fn	1099	682	857	399	185	15	1033	124	614	364	255	13
Vn	385	957	1399	1085	1172	1132	451	726	1361	1120	1176	1163
Acc(%)	43,88	53,10	63,26	82,57	80,61	85,69	36,66	65,63	71,26	82,45	78,07	86,97
)	%	%	%	%	%	%	%	%	%	%	%	%

Tabla 55. Resultados en HSI para la base de datos de videos exceptuando los videos de flama azul en la matriz de videos segmentados.

	BitXor						Diff					
	Cascade			Parallel			Cascade			Parallel		
	knn	nb	svm	knn	nb	svm	knn	nb	svm	knn	nb	svm
Vp	709	307	342	1025	992	1076	486	912	650	1023	1051	1075
Fp	371	263	9	55	314	301	594	612	31	57	336	262
Fn	1156	773	738	366	88	4	996	161	430	352	27	5
Vn	328	1217	1475	1118	1170	1183	488	865	1453	1132	1148	1222
Acc(%)	40,44	59,53	70,87	83,58	84,32	88,10	37,99	69,69	82,02	84,05	85,83	89,59
)	%	%	%	%	%	%	%	%	%	%	%	%

Tabla 56. Resultados en HSI para la base de datos de videos exceptuando los videos de flama azul en la matriz de videos originales y segmentados unidos.

	BitXor						Diff					
	Cascade			Parallel			Cascade			Parallel		
	knn	nb	svm	knn	nb	svm	knn	nb	svm	knn	nb	svm
Vp	737	395	359	1043	994	1076	481	983	673	1014	1029	1075
Fp	343	436	6	37	305	301	599	639	18	66	288	262
Fn	1101	685	721	366	84	4	1037	94	407	346	49	5
Vn	383	1032	1478	1118	1178	1183	447	830	1466	1138	1195	1222

Al eliminar los videos de flama azul no se mejoran los resultados, CMYK en video original con los 38 videos dio un porcentaje de precisión de 93.8% y sin los videos de flama azul 92.90%, CMYK en video segmentado con los 38 videos dio un porcentaje de precisión de 93.4% y sin los videos de flama azul 93.25% y CMYK en video original y segmentado con los 38 videos dio un porcentaje de precisión de 93.66% y sin los videos de flama azul 93.10%.

Con base a lo anterior y teniendo en cuenta la poca información encontrada, y aun que metodológicamente la textura no ofrece información relevante pero si es un método alternativo al movimiento. Como la interrelación del color no se puede utilizar por sí mismo se implementó para la detección de movimiento el método "Background Subtraction" el cual utiliza el movimiento y el color en un mismo procedimiento, sin embargo, a este método se le combinaron dos variantes utilizando el operador lógico XOR y otra con el valor absoluto de la diferencia de los frames. De esta manera en el método se contemplaron dos topologías de interconexión entre dos técnicas utilizadas para la detección de flamas (la detección de movimiento y el reconocimiento por color).

5. Conclusiones y recomendaciones

5.1 Conclusiones

- Al realizar el análisis comparativo de las técnicas de segmentación la topología en cascada no es la ideal para obtener mejores resultados.
- Al realizar el entrenamiento y validación la técnica que mejores resultados arroja es Bitxor en el espacio de color cmyk a través de la topología en paralelo. El espacio de color normalizado nos aporta mejores resultados que los no normalizados.
- Se logró extraer características medibles de las secuencias de imágenes para entrenar los clasificadores.
- Con un porcentaje de precisión del 93.1% se estableció un espacio de representación.
- Se logró extraer características medibles de las secuencias de imágenes para entrenar los clasificadores. Con un porcentaje de precisión del 93.1% se estableció un espacio de representación.

5.2 5.2 Recomendaciones

Se propone que, en trabajos futuros, se podrían mejorar los resultados de precisión del clasificador, usando más características medibles como el brillo y la luminiscencia, ajustando el contraste y de brillo de los frames y trabajando un espacio de color que sea resultado de la mezcla de algunas capas de los espacios de color, trabajados como Y, H y L.

Metodología para la identificación automática de flama/humo por medio de análisis de patrones dinámicos de imágenes digitales en entornos abiertos

A. Anexo: Código de la metodología

Examplesegmentatiofire.m

```
clc;clear all;
[filename, pathname] = ...
    uigetfile({'*.avi'; '*.mp4'}, 'File Selector');
Temp = [dir(fullfile(pathname, ['*.avi'])); ...
    dir(fullfile(pathname, ['*.mp4']))];
strColorSpace = {'rgb', 'ycbcr', 'cmy', 'cmyk', 'hsv', 'hsi', 'hsl'};
name = {}; Pixels = {};
opt = 'c';

if opt == 'p'
    subArchive = 'parallel';
elseif opt == 'c'
    subArchive = 'cascade';
end
strStore = fullfile('.\Mat1', 'SegmentationFire', subArchive);

for I = 1 : length(strColorSpace)
    for iArchive = 1 : length(Temp)
        strFilename = fullfile(pathname, [Temp(iArchive).name]);

        % Script que segmenta la imagen por color y movimiento
        [Resolution(iArchive,:), TempName, Diff, BitXor, ...
            ColorMask, DiffColorMask, ...
            BitXorColorMask, Original, iPixels] = ...
            SegmentationFire(strFilename, strColorSpace{I}, opt);

        name(iArchive) = {TempName};
        Pixels(iArchive) = {iPixels};

        % Se reconfiguran la informacion almacenada para generar una sola
        % estructura por cada video con con un arreglo por cada campo (caracteristica)
        staDiff(iArchive) = reconfigurationStruct(Diff); % Metodo Diff
        staBitXor(iArchive) = reconfigurationStruct(BitXor); % Metodo BitXor
        % Fusion Diff - Mascara de Color
```

```
staDiffColorMask(iArchive) = reconfigurationStruct(DiffColorMask);
% Fusion BitXor - Mascara de Color
staBitXorColorMask(iArchive) = reconfigurationStruct(BitXorColorMask);
% Datos de las imagenes originales
staOriginal(iArchive) = reconfigurationStruct(Original);

if strcmp(opt,'p')
    staColorMask(iArchive) = reconfigurationStruct(ColorMask); % Mascara de Color
else
    staColorMask = [];
end
save(fullfile(strStore,[opt 'SegmentationFire-' strColorSpace{1}
'.mat']),'Resolution','name', ...
    'staDiff','staBitXor','staColorMask','staDiffColorMask', ...
    'staBitXorColorMask','staOriginal','Pixels');
end
end
```

Segmentatiofire.m

```
% Script que detecta fuego a partir del movimiento y el color
function [Resolution,name,staDiff,staBitXor,staColorMask, ...
    staDiffMask,staBitXorMask,Original,Pixels,objMove] = ...
    SegmentationFire(strFilename,spacecolor,opt)

if nargin < 3,opt = 'p';end % Filtros en Paralelo

if opt == 'p'
    subArchive = 'parallel';
elseif opt == 'c'
    subArchive = 'cascade';
end
strStore = fullfile('\Mat1',mfilename);
strVideo = fullfile('\Videos1',mfilename,subArchive,spacecolor);

%% Procesamiento
objVideo = VideoReader(strFilename); % Abre el archivo de video

h = figure('NumberTitle','off','Name',['Detect Movement -> Frames: ' ...
    num2str(objVideo.NumberOfFrames)],'Color',[1 1 1]);
set(h,'Position',get(0,'Screensize')); % Maximize figure.
```

```

initX = 0.0499;initY = 0.6594;dx = 0.2137;dy = 0.2766;

Resolution = [objVideo.Height,objVideo.Width];
[pathstr,name,ext] = fileparts(strFilename);

% Estructura de Almacenamiento de la Informacion Segmentada
objMove = struct();
% Original = struct();
% staDiff = struct();
% staBitXor = struct();
% staColorMask = struct();
% staDiffMask = struct();
% staBitXorMask = struct();

% Condicional que permite determina si la ruta de almacenamiento existe
% y si no es el caso la crea
if exist(strVideo) == 0, mkdir(strVideo);end
% configuracion para la grabacion de los Frames
movie = VideoWriter(fullfile(strVideo,[mfilename '-' name '.mp4']));
movie.FrameRate = objVideo.FrameRate;
open(movie);

for Index = 1: objVideo.NumberOfFrames - 1
    %% Segmentacion x Movimiento
    % RGB Color Space
    img1 = read(objVideo, Index);
    objMove(Index).AllPixels = length(PixelsNoZero(img1));
    Img1 = im2double(img1);
    %img1 = normalizedSpaceColor(img1);
    img2 = im2double(read(objVideo, Index + 1));
    %img2 = normalizedSpaceColor(img2);

    % Method falsecolor
    Temp = imfuse(Img1,img2,'falsecolor');
    subplot('Position',[initX,initY,dx,dy]);
    imshow(Temp);
    title(['Movement Detect (Frame ' num2str(Index) ...
        ' de ' num2str(objVideo.NumberOfFrames) ')']);
    ylabel(strcat('False Color',['Pixels: ',...
        num2str(length(PixelsNoZero(Temp)))]));

    % Method diff
    Temp = im2double(imfuse(Img1,img2,'diff'));
    subplot('Position',[initX,initY-dy,dx,dy]);
    imshow(Temp);ylabel(strcat('Diff',['Pixels: ',...
        num2str(length(PixelsNoZero(Temp)))]));
    L1 = SelectRegions(Temp,graythresh(Temp)); % Imagen regionalizada
    subplot('Position',[initX+1.1*dx,initY-dy,dx,dy]);

```

```

imshow(L1);
[objMove(Index).Diff.Index,objMove(Index).Diff.Values] = ...
    PixelsNoZero(L1);
ylabel(['Pixels: ',num2str(length(objMove(Index).Diff.Index))]);
% Estadísticas de la imagen
staTemp = ImageStatisticsV1(Img1,L1);
Original(Index) = staTemp.Original;
L2 = PixelsNoZero(Img1);
Pixels(Index,1) = length(L2);
staDiff(Index) = staTemp.ColorSegmentation;
Pixels(Index,2) = length(objMove(Index).Diff.Index);

% Method bitxor
[L2,~,Temp] = BackgroundSubtraction(Img1,img2);
subplot('Position',[initX,initY-2*dy,dx,dy]);
imshow(Temp);ylabel(strvcat('BitXor',['Pixels: ',...
    num2str(length(PixelsNoZero(Temp)))]));
subplot('Position',[initX+1.1*dx,initY-2*dy,dx,dy]);
imshow(L2);
[objMove(Index).BitXor.Index,objMove(Index).BitXor.Values] = ...
    PixelsNoZero(L2);
ylabel(['Pixels: ',num2str(length(objMove(Index).BitXor.Index))]);
% Estadísticas de la imagen
staTemp = ImageStatisticsV1(Img1,L2);
staBitXor(Index) = staTemp.ColorSegmentation;
Pixels(Index,3) = length(objMove(Index).BitXor.Index);

%% Segmentacion x Color
if strcmp(opt,'p') % Parallel Filters
    ColorMask = ColorFilter(Img1,spacecolor);
    ColorMask = SelectRegions(ColorMask,0);
    ColorMask = double(ColorMask);
    ColorMask(find(ColorMask ~= 0)) = 1.0;
    [objMove(Index).ColorMask.Index,objMove(Index).ColorMask.Values] = ...
        PixelsNoZero(ColorMask);

% Estadísticas de la imagen
staTemp = ImageStatisticsV1(Img1,ColorMask);
staColorMask(Index) = staTemp.ColorSegmentation;
Pixels(Index,4) = length(objMove(Index).ColorMask.Index);

% Consolidación las regiones comunes a las dos máscaras
DiffMask = ConsolideRegion(L1,ColorMask);
BitXorMask = ConsolideRegion(L2,ColorMask);

subplot('Position',[initX+2.2*dx,initY-dy,dx,dy]);
imshow(ColorMask);
title(strvcat('Parallel Color Segmentation', ...
    [upper(spacecolor),' Color Space']));

```

```

        ylabel(['Pixels: ',num2str(length(objMove(Index).ColorMask.Index))]);
%     subplot('Position',[initX+2.2*dx,initY-dy,dx,dy]);
%     imshow(ColorMask);
%     subplot('Position',[initX+2.2*dx,initY-2*dy,dx,dy]);
%     imshow(ColorMask);
elseif strcmp(opt,'c') % Cascade Filters
    Temp = Mask3Layers(uint8(L1),img1);
%     subplot('Position',[initX+2.2*dx,initY-dy,dx,dy]);
%     imshow(Temp);
    [DiffMask,Img1] = ColorFilter(Temp,spacecolor);
    DiffMask = Mask3Layers(DiffMask,Img1);

    Temp = Mask3Layers(uint8(L2),img1);
%     subplot('Position',[initX+2.2*dx,initY-2*dy,dx,dy]);
%     imshow(Temp);
    [BitXorMask,img2] = ColorFilter(Temp,spacecolor);
    BitXorMask = Mask3Layers(BitXorMask,img2);
end
subplot('Position',[initX+3.3*dx,initY-dy,dx,dy]);
imshow(DiffMask);title('Result Segmentation');
[objMove(Index).ColorFilter.(spacecolor).Index, ...
 objMove(Index).ColorFilter.(spacecolor).Values] = ...
    PixelsNoZero(DiffMask);
ylabel(['Pixels: ',num2str(length(objMove(Index).ColorFilter.(spacecolor).Index))]);
% Estadísticas de la imagen
staTemp = ImageStatisticsV1(Img1,DiffMask);
staDiffMask(Index) = staTemp.ColorSegmentation;
Pixels(Index,5) = length(objMove(Index).ColorFilter.(spacecolor).Index);

subplot('Position',[initX+3.3*dx,initY-2*dy,dx,dy]);
imshow(BitXorMask);
[objMove(Index).ColorFilter.(spacecolor).Index, ...
 objMove(Index).ColorFilter.(spacecolor).Values] = ...
    PixelsNoZero(BitXorMask);
ylabel(['Pixels: ',num2str(length(objMove(Index).ColorFilter.(spacecolor).Index))]);
% Estadísticas de la imagen
staTemp = ImageStatisticsV1(Img1,BitXorMask);
staBitXorMask(Index) = staTemp.ColorSegmentation;
Pixels(Index,6) = length(objMove(Index).ColorFilter.(spacecolor).Index);

if(Index == 1)
    hax = findobj(gcf,'type','axes');

    for I = 1 : length(hax)
        set(hax(I),'NextPlot','replacechildren');
        hYLabel(I) = get(hax(I),'YLabel');
        pos = get(hYLabel(I),'Position');
        pos(1) = 0.2*pos(1);
    end
end

```

```
        set(hYLabel(l),'Position',pos);
    end
end
% Graba el frame en el Archivo de Video
f = getframe(h);
writeVideo(movie,f.cdata);
pause(0.01);
end
close(movie);
Temp = fullfile(strStore,subArchive,spacecolor);

% Condicional que permite determina si la ruta de almacenamiento existe
% y si no es el caso la crea
if exist(Temp) == 0, mkdir(Temp);end
% Archivo propietario donde se almacena la informacion de interes
save(fullfile(Temp,[filename '-' name '.mat']),'Resolution','name','objMove');

if strcmp(opt,'c') % Cascade Filters
    ColorMask = [];
    staColorMask = [];
end
```

Colorfilter.m

```
function [Mask,convert_image] = ColorFilter(img,space,opt)
    if nargin < 3,opt = '';end

    switch space
        case 'rgb'
            [Mask,convert_image] = filterrgb(img,opt);
        case 'cmy'
            [Mask,convert_image] = filtercmy(img,opt);
        case 'cmyk'
            [Mask,convert_image] = filtercmyk(img,opt);
```

```

    case 'ycbcr'
        [Mask,convert_image] = filterycbcr(img,0,opt);
    case 'hsv'
        [Mask,convert_image] = filterhsv(img,opt);
    case 'hsi'
        [Mask,convert_image] = filterhsi(img,opt);
    case 'hsl'
        [Mask,convert_image] = filterhsl(img,opt);
    otherwise
        return;
    end
end

function [Mask,rgbn_image] = filterrgb(img,opt)
    rgbn_image = im2double(img);

    % Normalizacion del espacio de color
    if find((opt == 'n') == 1)
        Temp = (rgbn_image(:,:,1).^2 + rgbn_image(:,:,2).^2 + rgbn_image(:,:,3).^2).^(1/2);
        rgbn_image(:,:,1) = rgbn_image(:,:,1)./Temp; % Espacio RGB Normalizado
        rgbn_image(:,:,2) = rgbn_image(:,:,2)./Temp;
        rgbn_image(:,:,3) = rgbn_image(:,:,3)./Temp;
    end
    % [m,n,~] = size(img);
    % BackGround = double(zeros(m,n));
    % % Rule 1 -> R>G>B
    % Mask = BackGround;
    % for Row = 1:m
    %     for Col = 1:n
    %         if((img(Row,Col,1)>img(Row,Col,2)) && (img(Row,Col,2)>img(Row,Col,3)))
    %             Mask(Row,Col) = 1.0;
    %         end
    %     end
    % end
    % end
    Mask = double((rgbn_image(:,:,1) >
    rgbn_image(:,:,2)).*(rgbn_image(:,:,2)>rgbn_image(:,:,3)));

    if find((opt == 'z') == 1)
        Index = find(Mask ~= 0);
        Temp = rgbn_image(:,:,1);
        %Rmean = mean2(Temp(Index));
        Rmean = mean(mean(Temp(Index)));
        Temp = rgbn_image(:,:,2);
        %Gmean = mean2(Temp(Index));
        Gmean = mean(mean(Temp(Index)));
        Temp = rgbn_image(:,:,3);
        %Bmean = mean2(Temp(Index));
        Bmean = mean(mean(Temp(Index)));
    end
end

```



```

else
    Rmean = mean(mean(img(:,:,1)));
    Gmean = mean(mean(img(:,:,2)));
    Bmean = mean(mean(img(:,:,3)));
end
RTemp = (rgb_image(:,:,1) > Rmean);
GTemp = (rgb_image(:,:,2) > Gmean);
BTemp = (rgb_image(:,:,3) < Bmean);

if isempty(find(Mask ~= 0))
    Mask = double((RTemp.*GTemp).*BTemp);
else
    Mask = Mask .* double((RTemp.*GTemp).*BTemp);
end
end

function [Temp,temp_image] = filtercmy(img,opt)
temp_image = rgb2cmyk(im2double(img));
temp_image = temp_image(:,:,1:3);

% Normalizacion del espacio de color
if find((opt == 'n') == 1)
    Temp = temp_image(:,:,1) + temp_image(:,:,2) + temp_image(:,:,3);
    temp_image(:,:,1) = temp_image(:,:,1)./Temp;
    temp_image(:,:,2) = temp_image(:,:,2)./Temp;
    temp_image(:,:,3) = temp_image(:,:,3)./Temp;
end
[m,n,~] = size(img);
BackGround = zeros(m,n);
Temp = BackGround;

% Rule 1 -> C > M > Y
for Row = 1:m
    for Col = 1:n
        if((temp_image(Row,Col,1) < temp_image(Row,Col,2)) &&
(temp_image(Row,Col,2) < temp_image(Row,Col,3)))
            Temp(Row,Col) = 1;
        end
    end
end

% Rule 2 -> C < Cmean && M < Mmean && Y > Ymean
if find((opt == 'z') == 1)
    Index = find(Temp ~= 0);
    Cmean = temp_image(:,:,1);
    Cmean = mean(mean(Cmean(Index)));
    Mmean = temp_image(:,:,2);
    Mmean = mean(mean(Mmean(Index)));
    Ymean = temp_image(:,:,3);

```

```

    Ymean = mean(mean(Ymean(Index)));
else
    Cmean = mean(mean(temp_image(:,:,1)));
    Mmean = mean(mean(temp_image(:,:,2)));
    Ymean = mean(mean(temp_image(:,:,3)));
end

% Capa Cyan
Temp = Temp .* (temp_image(:,:,1) <= Cmean);
% Capa Magenta
Temp = Temp .* (temp_image(:,:,2) <= Mmean);
% Capa Yellow
Temp = Temp .* (temp_image(:,:,3) >= Ymean);
Temp = double(Temp);
end

function [Temp,temp_image] = filtercmyk(img,opt)
temp_image = rgb2cmyk(im2double(img));
temp_image = temp_image(:,:,1:3);

% Normalizacion del espacio de color
if find((opt == 'n') == 1)
    Temp = temp_image(:,:,1) + temp_image(:,:,2) + temp_image(:,:,3);
    temp_image(:,:,1) = temp_image(:,:,1)./Temp;
    temp_image(:,:,2) = temp_image(:,:,2)./Temp;
    temp_image(:,:,3) = temp_image(:,:,3)./Temp;
end
[m,n,~] = size(img);
BackGround = zeros(m,n);
Temp = BackGround;

% Rule 1 -> C > M > Y
for Row = 1:m
    for Col = 1:n
        if((temp_image(Row,Col,1) < temp_image(Row,Col,2)) &&
(temp_image(Row,Col,2) < temp_image(Row,Col,3)))
            Temp(Row,Col) = 1;
        end
    end
end
end

% Rule 2 -> C < Cmean && M < Mmean && Y > Ymean
if find((opt == 'z') == 1)
    Index = find(Temp ~= 0);
    Cmean = temp_image(:,:,1);
    Cmean = mean(mean(Cmean(Index)));
    Mmean = temp_image(:,:,2);
    Mmean = mean(mean(Mmean(Index)));

```

```

    Ymean = temp_image(:,:,3);
    Ymean = mean(mean(Ymean(Index)));
else
    Cmean = mean(mean(temp_image(:,:,1)));
    Mmean = mean(mean(temp_image(:,:,2)));
    Ymean = mean(mean(temp_image(:,:,3)));
end

% Capa Cyan
Temp = Temp .* (temp_image(:,:,1) <= Cmean);
% Capa Magenta
Temp = Temp .* (temp_image(:,:,2) > Mmean);
% Capa Yellow
Temp = Temp .* (temp_image(:,:,3) >= Ymean);
Temp = double(Temp);
end

function [Mask,ycbcr_image] = filterycbcr(convert_image,th,opt)
    ycbcr_image = im2double(rgb2ycbcr(convert_image));

% Normalizacion del espacio de color
if find((opt == 'n') == 1)
    Mask = ycbcr_image(:,:,1) + ycbcr_image(:,:,2) + ycbcr_image(:,:,3);
    ycbcr_image(:,:,1) = ycbcr_image(:,:,1)./Mask;
    ycbcr_image(:,:,2) = ycbcr_image(:,:,2)./Mask;
    ycbcr_image(:,:,3) = ycbcr_image(:,:,3)./Mask;
end

% Rule 1 -> Y >= Cb
Mask = (ycbcr_image(:,:,1) >= ycbcr_image(:,:,2));

% Rule 2 -> Cr >= Cb
Temp = (ycbcr_image(:,:,3) >= ycbcr_image(:,:,2));
Mask = Mask + Temp;

% Rule 3 -> Y >= Ymean && Cb <= Cbmean && Cr >= Crmean
if find((opt == 'z') == 1)
    Temp = ycbcr_image(:,:,1);
    YTemp = (ycbcr_image(:,:,1) >= mean(mean(Temp(find(Mask ~= 0)))));
    Temp = ycbcr_image(:,:,2);
    CbTemp = (ycbcr_image(:,:,2) <= mean(mean(Temp(find(Mask ~= 0)))));
    Temp = ycbcr_image(:,:,3);
    CrTemp = (ycbcr_image(:,:,3) >= mean(mean(Temp(find(Mask ~= 0)))));
else
    YTemp = (ycbcr_image(:,:,1) >= mean(mean(ycbcr_image(:,:,1))));
    CbTemp = (ycbcr_image(:,:,2) <= mean(mean(ycbcr_image(:,:,2))));
    CrTemp = (ycbcr_image(:,:,3) >= mean(mean(ycbcr_image(:,:,3))));
end
Temp = ((YTemp.*CbTemp).*CrTemp);

```

```

Mask = Mask + Temp;

% Rule 4 -> Cb - Cr >= Th
Temp = (abs(ycbcr_image(:,:,2)-ycbcr_image(:,:,2)) >= th);
Mask = Mask.*Temp;

% Rule 5 -> (Cb <= 120) && (Cr >= 150)
[m,n,~] = size(ycbcr_image);
BackGround = (zeros(m,n));
CbTemp = find(ycbcr_image(:,:,2) <= 0.4706);
Temp = BackGround;
if ~isempty(CbTemp),Temp(CbTemp) = 1.0;end

CrTemp = find(ycbcr_image(:,:,3) >= 0.5882);
YTemp = BackGround;
if ~isempty(CrTemp),YTemp(CrTemp) = 1.0;end
Temp = (Temp + YTemp);

Mask = double(Mask .* Temp);
end

function [Temp,temp_img] = filterhsv(img,opt)
temp_img = rgb2hsv(img); % conversion de RGB a HSI

if find((opt == 'n') == 1)
    Temp = temp_img(:,:,1) + temp_img(:,:,2) + temp_img(:,:,3);
    temp_img(:,:,1) = temp_img(:,:,1)./Temp;
    temp_img(:,:,2) = temp_img(:,:,2)./Temp;
    temp_img(:,:,3) = temp_img(:,:,3)./Temp;
end

if find((opt == 'z') == 1)
    % Hue Layer
    Mask = temp_img(:,:,1);
    level(1) = graythresh(Mask(find(Mask ~= 0)));
    % Saturation Layer
    Mask = temp_img(:,:,2);
    level(2) = graythresh(Mask(find(Mask ~= 0)));
    % Brightness Layer
    Mask = temp_img(:,:,3);
    level(3) = graythresh(Mask(find(Mask ~= 0)));
else
    % Hue Layer
    level(1) = graythresh(temp_img(:,:,1));
    % Saturation Layer
    level(2) = graythresh(temp_img(:,:,2));
    % Brightness Layer
    level(3) = graythresh(temp_img(:,:,3));
end

```

```

end
% Hue Layer
Temp(:,:,1) = temp_img(:,:,1) < level(1);
% Saturation Layer
Temp(:,:,2) = temp_img(:,:,2) > level(2);
% Value Layer
Temp(:,:,3) = temp_img(:,:,3) > level(3);

% Mascara Definitiva de Filtrado
Temp = double(Temp(:,:,1).*Temp(:,:,2).*Temp(:,:,3));
Temp(find(Temp ~= 0.0)) = 1.0;
end

function [Temp,temp_img] = filterhsl(img,opt)
temp_img = rgb2hsl(im2double(img)); % conversion de RGB a HSI

if find((opt == 'n') == 1)
    Temp = temp_img(:,:,1) + temp_img(:,:,2) + temp_img(:,:,3);
    temp_img(:,:,1) = temp_img(:,:,1)./Temp;
    temp_img(:,:,2) = temp_img(:,:,2)./Temp;
    temp_img(:,:,3) = temp_img(:,:,3)./Temp;
end

if find((opt == 'z') == 1)
    % Hue Layer
    Mask = temp_img(:,:,1);
    level(1) = graythresh(Mask(find(Mask ~= 0)));
    % Saturation Layer
    Mask = temp_img(:,:,2);
    level(2) = graythresh(Mask(find(Mask ~= 0)));
    % Luminescence Layer
    Mask = temp_img(:,:,3);
    level(3) = graythresh(Mask(find(Mask ~= 0)));
else
    % Hue Layer
    level(1) = graythresh(temp_img(:,:,1));
    % Saturation Layer
    level(2) = graythresh(temp_img(:,:,2));
    % Luminescence Layer
    level(3) = graythresh(temp_img(:,:,3));
end
end
% Hue Layer
Temp(:,:,1) = temp_img(:,:,1) < level(1);
% Saturation Layer
Temp(:,:,2) = temp_img(:,:,2) > level(2)/2;
% Intensity Layer
Temp(:,:,3) = temp_img(:,:,3) > level(3);

% Mascara Definitiva de Filtrado

```

```

Temp = double(Temp(:,:,1).*Temp(:,:,2).*Temp(:,:,3));
Temp(find(Temp ~= 0.0)) = 1.0;
end

function [Temp,temp_img] = filterhsi(img,opt)
temp_img = abs(rgb2hsi(im2double(img))); % conversion de RGB a HSI

if find((opt == 'n') == 1)
    Temp = temp_img(:,:,1) + temp_img(:,:,2) + temp_img(:,:,3);
    temp_img(:,:,1) = temp_img(:,:,1)./Temp;
    temp_img(:,:,2) = temp_img(:,:,2)./Temp;
    temp_img(:,:,3) = temp_img(:,:,3)./Temp;
end

if find((opt == 'z') == 1)
    % Hue Layer
    Mask = temp_img(:,:,1);
    level(1) = graythresh(Mask(find(Mask ~= 0)));
    % Saturation Layer
    Mask = temp_img(:,:,2);
    level(2) = graythresh(Mask(find(Mask ~= 0)));
    % Intensity Layer
    Mask = temp_img(:,:,3);
    level(3) = graythresh(Mask(find(Mask ~= 0)));
else
    % Hue Layer
    level(1) = graythresh(temp_img(:,:,1));
    % Saturation Layer
    level(2) = graythresh(temp_img(:,:,2));
    % Intensity Layer
    level(3) = graythresh(temp_img(:,:,3));
end
% Hue Layer
Temp(:,:,1) = temp_img(:,:,1) < level(1);
% Saturation Layer
Temp(:,:,2) = temp_img(:,:,2) > level(2);
% Intensity Layer
Temp(:,:,3) = temp_img(:,:,3) > level(3);

% Mascara Definitiva de Filtrado
Temp = double(Temp(:,:,1).*Temp(:,:,2).*Temp(:,:,3));
Temp(find(Temp ~= 0.0)) = 1.0;
end

```

SelectRegions.m

```
% Script que permite etiquetar y filtrar algunas de las regiones generadas
% a partir de la segmentacion de la imagen
function [L2,FilteredImage] = SelectRegions(img,level)
    %level = graythresh(img);
    FilteredImage = img > level; % Binarizacion de la Imagen
    %Apply Median filter to remove Noise
    FilteredImage = medfilt2(FilteredImage,[5 5]);

    %Boundary Label the Filtered Image
    [L num] = bwlabel(FilteredImage);
    STATS = regionprops(L,'all');
    cc = [];
    removed = 0;

    %Remove the noisy regions
    for i = 1:num
        dd = STATS(i).Area;

        if (dd < 0.2*dd)
            L(L == i) = 0;
            removed = removed + 1;
            num = num-1;
        end
    end
    [L2 num2] = bwlabel(L);
```

Rgb2cmy.m

```
function [c,m,y] = rgb2cmy(img)
    if strcmp(class(img),'uint8')
        c = 255-img(:,:,1);
        m = 255-img(:,:,2);
        y = 255-img(:,:,3);
    elseif strcmp(class(img),'double')
        c = 1.0-img(:,:,1);
        m = 1.0-img(:,:,2);
        y = 1.0-img(:,:,3);
    end

    if nargin < 2
        c = cat(3,c,m,y);
    end
```


Rgb2cmyk.m

```
function c = rgb2cmyk(c)

% RGB2CMYK  Converts between RGB- and CMYK- Colors
% RedGreenBlue <--> CyanMagentaYellowBlack
% CMYK = RGB2CMYK( RGB )
% RGB = RGB2CMYK( CMYK )
% RGB : N by 3 , N Colors
% CMYK: N by 4
% I found some "generic" conversion formula between RGB and CMYK (both are 0-1
based):
% Black = minimum(1-Red,1-Green,1-Blue)
% Cyan = (1-Red-Black)/(1-Black)
% Magenta = (1-Green-Black)/(1-Black)
% Yellow = (1-Blue-Black)/(1-Black)

s = size(c);
n = size(s,2);

m = s(2+(n==3));

if ~( isnumeric(c) & any( n == [ 2 3 ] ) & ...
    any( m == [ 3 4 ] ) )
    error('Input must be a RGB or CMYK ColorMatrice.');
```

```
if isempty(c)
    return
end

u8 = strcmp(class(c),'uint8');
if u8
    c = double(c)/255;
end

sub = { ':' ':' };
sub = sub(1:(1+(n==3)));

if m == 3
    % RGB --> CMYK

    c = 1 - c;
    k = min(c,[],n);

    c = c - k(sub{:},[1 1 1]);

    c = cat( n , c , k );

else
    % CMYK --> RGB
    c = 1 - ( c(sub{:},[1 2 3]) + c(sub{:},[4 4 4]) );

end

if u8
    c = uint8(round(c*255));
end
```

Rgb2hsi.m

```
function [H,S,I] = rgb2hsi(img)
    R = img(:,:,1);G = img(:,:,2);B = img(:,:,3);
    I = (R+G+B)/3;
    Hnum = ((R-G) + (R-B));
    Hden = 2*sqrt(double((R-G).^2 + (R-B).*(G-B)));
    H = acos(double(Hnum)./double(Hden));
    S = 1 -3*min(img,[],3)./I;

    if strcmp(class(img),'uint8')
        I = I/255;
    end
```

```
if nargin < 2
    H = cat(3,H,S,I);
end
```

Rgb2hsl.m

```
function hsl=rgb2hsl(rgb_in)
%Converts Red-Green-Blue Color value to Hue-Saturation-Luminance Color value
%
%Usage
%    HSL = rgb2hsl(RGB)
```

```
%
% converts RGB, a M [x N] x 3 color matrix with values between 0 and 1
% into HSL, a M [x N] X 3 color matrix with values between 0 and 1
%
%See also hsl2rgb, rgb2hsv, hsv2rgb

% (C) Vladimir Bychkovsky, June 2008
% written using:
% - an implementation by Suresh E Joel, April 26,2003
% - Wikipedia: http://en.wikipedia.org/wiki/HSL\_and\_HSV

rgb = reshape(rgb_in, [], 3);

mx = max(rgb,[],2);%max of the 3 colors
mn = min(rgb,[],2);%min of the 3 colors

L = (mx+mn)/2;%luminance is half of max value + min value
S = zeros(size(L));

% this set of matrix operations can probably be done as an addition...
zeroidx = (mx==mn);
S(zeroidx) = 0;

lowlidx =L <= 0.5;
calc = (mx-mn)./(mx+mn);
idx = lowlidx & (~ zeroidx);
S(idx) = calc(idx);

hilidx = L > 0.5;
calc = (mx-mn)./(2-(mx+mn));
idx = hilidx & (~ zeroidx);
S(idx) = calc(idx);

hsv = rgb2hsv(rgb);
H = hsv(:,1);

hsl = [H, S, L];

hsl = round(hsl.*100000)./100000;
hsl = reshape(hsl, size(rgb_in));
```

rgb2ycbcr.m

```
function [Y,Cb,Cr] = rgb2ycbcr(img)
    if strcmp(class(img),'double')
```

```
    img = uint8(255*img);  
end  
R = img(:,:,1);G = img(:,:,2);B = img(:,:,3);  
Y = uint8(0.2568*R+0.5041*G+0.0979*B) + 16;  
Cb = uint8(-0.1482*R-0.2910*G+0.4392*B) + 128;  
Cr = uint8(0.4392*R-0.3678*G-0.0714*B) + 128;  
  
if nargout < 2  
    Y = cat(3,Y,Cb,Cr);  
end  
end
```

Reconfigurationstruct.m

```
% Script que permite convertir una matriz de estructuras en una estructura  
% con matrices
```

```
function staDiff = reconfigurationStruct(Diff)  
    fn = fieldnames(Diff);
```

```
% Ciclo que recorre todos los campos de la estructura
```

```
for iField = 1 : length(fn)
```

```
    iTemp = [];
```

```
% Ciclo que recorre toda la matriz de estructuras
```

```
for lter = 1 : length(Diff)
```

```
    iTemp = [iTemp;Diff(lter).(fn{iField})];
```

```
end
```

```
    staDiff.(fn{iField}) = iTemp;
```

```
end
```

```
end
```

Selectregions.m

% Script que permite etiquetar y filtrar algunas de las regiones generadas
% a partir de la segmentacion de la imagen

```
function [L2,FilteredImage] = SelectRegions(img,level)
    %level = graythresh(img);
    FilteredImage = img > level;    % Binarizacion de la Imagen
    %Apply Median filter to remove Noise
    FilteredImage = medfilt2(FilteredImage,[5 5]);
```

%Boundary Label the Filtered Image

```
[L num] = bwlabel(FilteredImage);
STATS = regionprops(L,'all');
cc = [];
removed = 0;
```

%Remove the noisy regions

```
for i = 1:num
    dd = STATS(i).Area;

    if (dd < 0.2*dd)
        L(L == i) = 0;
        removed = removed + 1;
        num = num-1;
    end
end
[L2 num2] = bwlabel(L);
```


ImageStatisticsV1.m

```
% Script que permite extraer las características de los diferentes frames  
% Versiion 1.0
```

```
function statistics = ImageStatisticsV1(img,mask)  
    [m,n,Layers] = size(img);  
    strField = {'Original','ColorSegmentation'};  
  
    for lter = 1 : length(strField)  
        for l = 1 : Layers % Calculo x Cada Capa de Color Base  
            if(lter == 1) % Imagen Original  
                % Organizado x Filas  
                x(:,l) = double(reshape(img(:,:,1),[],1));  
                % Organizado x Columnas  
                xTemp = im2double(img);  
                x = reshape(im2double(img(:,:,l)),[],1);  
            else  
                % Organizado x Filas  
                x(:,l) = double(reshape(img(:,:,1),[],1));  
                % Organizado x Columnas  
                xTemp = Mask3Layers(mask(:,:,1),im2double(xTemp));  
                x = reshape(im2double(img(:,:,l)).*mask(:,:,1),[],1);  
            end  
  
            % Promedio Total de los datos de la imagen original  
            statistics.(strField{lter}).mean(1,l) = mean(x);  
  
            % Mediana Total de los datos de la imagen original  
            statistics.(strField{lter}).median(1,l) = median(x);  
        end  
    end
```

```

% Moda Total de los datos de la imagen original
statistics.(strField{lter}).mode(1,l) = mode(x);

% Varianza Total de los datos de la imagen original
statistics.(strField{lter}).var(1,l) = var(x);

% Covarianza Total de los datos de la imagen
statistics.(strField{lter}).cov(1,l) = cov(x);

% Desviacion Estandar Total de los datos de la imagen original
statistics.(strField{lter}).std(1,l) = std(x);

% Coeficiente de Correlacion Total de los datos de la imagen
statistics.(strField{lter}).corrcoef(1,l) = corrcoef(x);

% Geometric mean Total de los datos de la imagen
statistics.(strField{lter}).geomean(1,l) = geomean(x);

% Harmonic mean Total de los datos de la imagen
statistics.(strField{lter}).harmmean(1,l) = harmmean(x);

% Mean or median absolute deviation Total de los datos de la imagen
statistics.(strField{lter}).mad(1,l) = mad(x);

% Calculo del coeficiente de apuntamiento de Fisher
statistics.(strField{lter}).kurtosis(1,l) = kurtosis(x);

% Calculo del coeficiente de asimetria de Fisher
statistics.(strField{lter}).skewness(1,l) = skewness(x);

% Calcula la probabilidad de que se presenten cada uno de los niveles
% color en cada capa (Frecuencia relativa)
XValue = (0:1:255);
probX = hist(x,XValue)/(m*n);
% Calculo de la Entropia
statistics.(strField{lter}).entropia(1,l) = entropy(probX);
statistics.(strField{lter}).entropia_shannon(1,l) = wentropy(probX, 'shannon', 0);

%% Valores estadisticos excluyendo los valores Nulos "0"
Index = find(x ~= 0);
% Promedio Total de los datos de la imagen original sin los "0"
statistics.(strField{lter}).meanNoZero(1,l) = mean(x(Index));

% Mediana Total de los datos de la imagen original
statistics.(strField{lter}).medianNoZero(1,l) = median(x(Index));

% Moda Total de los datos de la imagen original
statistics.(strField{lter}).modeNoZero(1,l) = mode(x(Index));

```

```
% Varianza Total de los datos de la imagen original sin los "0"
statistics.(strField{lter}).varNoZero(1,l) = var(x(Index));

% Covarianza Total de los datos de la imagen
statistics.(strField{lter}).covNoZero(1,l) = cov(x(Index));

% Desviacion Estandar Total de los datos de la imagen original sin los "0"
statistics.(strField{lter}).stdNoZero(1,l) = std(x(Index));

% Coeficiente de Correlacion Total de los datos de la imagen
statistics.(strField{lter}).corrcoefNoZero(1,l) = corrcoef(x(Index));

% Geometric mean Total de los datos de la imagen
statistics.(strField{lter}).geomeanNoZero(1,l) = geomean(x(Index));

% Harmonic mean Total de los datos de la imagen
statistics.(strField{lter}).harmmeanNoZero(1,l) = harmmean(x(Index));

% Mean or median absolute deviation Total de los datos de la imagen
statistics.(strField{lter}).madNoZero(1,l) = mad(x(Index));

% Calculo del coeficiente de apuntamiento de Fisher
statistics.(strField{lter}).kurtosisNoZero(1,l) = kurtosis(x(Index));

% Calculo del coeficiente de asimetria de Fisher
statistics.(strField{lter}).skewnessNoZero(1,l) = skewness(x(Index));

% Calculo de la Entropia
probX = hist(x(Index),XValue)/(m*n);
statistics.(strField{lter}).entropiaNoZero(1,l) = entropy(probX);
statistics.(strField{lter}).entropia_shannonNoZero(1,l) = wentropy(probX,
'shannon', 0);
    end
end
```

BackgroundSubtraction.m

```
function [L2,CurrentFrame_hsv,Out] = BackgroundSubtraction(Img1,Img2)
    %Convert RGB 2 HSV Color conversion
    [Background_hsv] = round(rgb2hsv(Img1));
    [CurrentFrame_hsv] = round(rgb2hsv(Img2));

    Out = bitxor(Background_hsv,CurrentFrame_hsv);
    %Convert RGB 2 GRAY
    Out = rgb2gray(Out);

    % Regionalization of the image
    %[L2,FilteredImage] = SelectRegions(Out,0);
    [L2,FilteredImage] = SelectRegions(Out,graythresh(Out));
end
```

Consolideregion.m

```
function MaskDiff = ConsolideRegion(L1,ColorMask)
MaskDiff = ColorMask .* L1;
```

```
Regions = unique(L1(find(MaskDiff ~= 0)));
Temp = zeros(size(MaskDiff));
```

```
% Se redimensiona la mascara con los pixeles de color que cumplen con
% la mascara de deteccion de movimiento
```

```
Regions = unique(ColorMask(find(MaskDiff ~= 0)));  
for l = 1 : length(Regions)  
    Temp(find(ColorMask == Regions(l))) = 1.0;  
end
```

B. Bibliografía

- B. Ugur Toreyin, Yigithan Dedeoglu, and A. E. C. (n.d.). Contour Based Smoke Detection In Video Using Wavelets. *European Commission 6th Frame- Work Programme under the Grant Number FP6-507752*, (iii).
- Bandi, M. A., & Patil, V. V. (2016). Fire Detection by using Digital Image Processing Technique, *5*(10), 2015–2017.
- Basset Blesa, J. M. (2000a). Flashover: Desarrollo y Control, 79. Retrieved from <http://www.olerdola.org/documentos/basset2.pdf>
- Basset Blesa, J. M. (2000b). Flashover: Desarrollo y Control, 79.
- Bharathi, S., Gokilapriya, S., Elango, N., & Vidhya, P. (2016). FIRE DETECTION AND FIRE SIGNATURE USING COLOR MODELS FOR SECURITY. *International Journal of Current Research and Modern Education (IJCRME)*, 79–88.
- Binti Zaidi, N. I., Binti Lokman, N. A. A., Bin Daud, M. R., Achmad, H., & Chia, K. A. (2015). Fire recognition using RGB and YCbCr color space. *ARPN Journal of Engineering and Applied Sciences*, *10*(21), 9786–9790.
- Buemi, A., Giacalone, D., Naccari, F., & Spampinato, G. (2016). Efficient Fire Detection Using Fuzzy Logic, 1–4.
- Celik, T., Demirel, H., Ozkaramanli, H., & Uyguroglu, M. (2007). Fire detection using statistical color model in video sequences. *Journal of Visual Communication and Image Representation*, *18*(2), 176–185. <https://doi.org/10.1016/j.jvcir.2006.12.003>
- Çetin, A. E. (2016). *METHODS AND TECHNIQUES FOR FIRE DETECTION Signal, Image and Video Processing Perspectives*. Elsevier.
- Çetin, a. E., Dimitropoulos, K., Gouverneur, B., Grammalidis, N., Günay, O., Habiboğlu, Y. H., ... Verstockt, S. (2013). Video fire detection – Review. *Digital Signal*

- Processing*, 23(6), 1827–1843. <https://doi.org/10.1016/j.dsp.2013.07.003>
- Chen, J., He, Y., & Wang, J. (2010). Multi-feature fusion based fast video flame detection. *Building and Environment*, 45(5), 1113–1122. <https://doi.org/10.1016/j.buildenv.2009.10.017>
- Chen, S.-J., Hovde, D. C., Peterson, K. a., & Marshall, A. W. (2007). Fire detection using smoke and gas sensors. *Fire Safety Journal*, 42(8), 507–515. <https://doi.org/10.1016/j.firesaf.2007.01.006>
- Chernick, M. ., Murthy, V. ., & Nealy, C. . (1985). Application of bootstrap and other resampling techniques: Evaluation of classifier performance. *Pattern Recognition Letters*, 3(3), 167–178. [https://doi.org/10.1016/0167-8655\(85\)90049-2](https://doi.org/10.1016/0167-8655(85)90049-2)
- Chino, D. Y. T., Avalhais, L. P. S., Rodrigues, J. F., & Traina, A. J. M. (2015). BoWFire: Detection of Fire in Still Images by Integrating Pixel Color and Texture Analysis. *Brazilian Symposium of Computer Graphic and Image Processing, 2015–Octob*, 95–102. <https://doi.org/10.1109/SIBGRAPI.2015.19>
- Delgado, M., Ruiz, M. D., Sánchez, D., & Vila, M. A. (2014). Fuzzy quantification: a state of the art. *Fuzzy Sets and Systems*, 242, 1–30. <https://doi.org/10.1016/j.fss.2013.10.012>
- Derbel, F. (2004). Performance improvement of fire detectors by means of gas sensors and neural networks. *Fire Safety Journal*, 39(5), 383–398. <https://doi.org/10.1016/j.firesaf.2004.03.001>
- En, S., Para, V., & Gratis, D. (n.d.). Libros univeristarios y solucionarios de muchos de estos libros gratis en descarga directa.
- Engineering, I. (2015). A Review on Flame and Smoke Detection Techniques in Video ' s. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, 855–859.
- Filizzola, C., Corrado, R., Marchese, F., Mazzeo, G., Paciello, R., Pergola, N., & Tramutoli, V. (2016). RST-FIRES, an exportable algorithm for early-fire detection and monitoring: description, implementation, and field validation in the case of the MSG-SEVIRI sensor. *Remote Sensing of Environment*, 186, 196–216. <https://doi.org/10.1016/j.rse.2016.08.008>
- Foggia, P., Saggese, A., & Vento, M. (2015). Real-Time Fire Detection for Video-Surveillance Applications Using a Combination of Experts Based on Color , Shape , and Motion. *TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY*, 25(9), 1545–1556.
- Frizzi, S., & Kaabi, R. (2016). Convolutional Neural Network for Video Fire and Smoke Detection, 877–882.
- Gonzalez, R. C. (2002). *Digital Imagen Processing*. (P. P. Hall, Ed.) (Third Edit). <https://doi.org/10:0-8400-5444-0>
- Gubbi, J., Marusic, S., & Palaniswami, M. (2009). Smoke detection in video using wavelets and support vector machines. *Fire Safety Journal*, 44(8), 1110–1115. <https://doi.org/10.1016/j.firesaf.2009.08.003>

- Gubernov, V., Kolobov, a., Polezhaev, a., Sidhu, H., & Mercer, G. (2010). Period doubling and chaotic transient in a model of chain-branching combustion wave propagation. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 466(2121), 2747–2769. <https://doi.org/10.1098/rspa.2009.0668>
- Han, J., & Bhanu, B. (2007). Fusion of color and infrared video for moving human detection. *Pattern Recognition*, 40(6), 1771–1784. <https://doi.org/10.1016/j.patcog.2006.11.010>
- Jiayun, W. (2015). Real-time Visual Detection of Early Manmade Fire, (August), 993–997.
- Kallur, R., & Kallur, P. R. (2016). Fire Detection With Wireless Video Camera. *International Journal of Recent Trends in Engineering & Research*, 316–321.
- Kandil, M., Salama, M., & Rashad, S. (2010). Fire Detection Using a Dynamically Developed Neural Network, (September), 15–17.
- Khatami, A., Mirghasemi, S., Khosravi, A., Lim, C. P., & Nahavandi, S. (2017). A new PSO-based approach to fire flame detection using K-Medoids clustering. *Expert Systems with Applications*, 68, 69–80. <https://doi.org/10.1016/j.eswa.2016.09.021>
- Ko, B., Cheong, K.-H., & Nam, J.-Y. (2010). Early fire detection algorithm based on irregular patterns of flames and hierarchical Bayesian Networks. *Fire Safety Journal*, 45(4), 262–270. <https://doi.org/10.1016/j.firesaf.2010.04.001>
- Ko, B., Jung, J.-H., & Nam, J.-Y. (2014). Fire detection and 3D surface reconstruction based on stereoscopic pictures and probabilistic fuzzy logic. *Fire Safety Journal*, 68, 61–70. <https://doi.org/10.1016/j.firesaf.2014.05.015>
- Ko, T., Soatto, S., & Estrin, D. (2008). LNCS 5304 - Background Subtraction on Distributions, 276–289.
- Kong, H. (2010). FIRE AND SMOKE DETECTION IN VIDEO WITH OPTIMAL MASS TRANSPORT BASED OPTICAL FLOW AND NEURAL NETWORKS Georgia Institute of Technology Schools of Electrical and Computer and Biomedical Engineering Atlanta , GA The University of British Columbia Dept . of Ma, 761–764.
- Kong, S. G., Jin, D., Li, S., & Kim, H. (2016). Fast fire flame detection in surveillance video using logistic regression and temporal smoothing. *Fire Safety Journal*, 79, 37–43. <https://doi.org/10.1016/j.firesaf.2015.11.015>
- Kurup, A. R. (2014). Vision Based Fire Flame Detection System Using Optical flow Features and Artificial Neural Network. *International Journal of Science and Research (IJSR)*, 3(10), 2161–2168.
- Luo, S., Yan, C., Wu, K., & Zheng, J. (2015). Smoke detection based on condensed image. *Fire Safety Journal*, 75, 23–35. <https://doi.org/10.1016/j.firesaf.2015.04.002>
- Lute, S., Sadaphal, A., Samudra, A., & Narkhede, A. (n.d.). Survey Paper of Approaches for Real Time Fire Detection, 6258–6261.
- Mathi, P. T., & Latha, L. (2016). Video Based Forest Fire Detection using Spatio-Temporal Flame Modeling and Dynamic Texture Analysis, 2(4), 41–47.

- Mishra, D., Yadav, A., Ray, S., & Kalra, P. K. (2007). Artificial neural network type learning with single multiplicative spiking neuron. *International Journal*, 8(1), 29–41.
- Oreyin, B. cet U. gur T. (2009). FIRE DETECTION ALGORITHMS USING MULTIMODAL SIGNAL AND IMAGE ANALYSIS.
- Paper, C., Poobalan, K., Universiti, S. L., & Liew, S. (2015). Fire Detection Algorithm Using Image, (December), 12–13.
- Pevzner, P., & Waterman, M. (2009). *Lecture Notes in Bioinformatics*.
- Piccardi, M. (2004). Background Subtraction Techniques: A Review. *IEEE International Conference on Systems, Man, and Cybernetics*, 4, 3099–3104. <https://doi.org/10.1109/ICSMC.2004.1400815>
- Piccinini, P., Calderara, S., & Cucchiara, R. (2008). Reliable smoke detection system in the domains of image energy and color, 1376–1379.
- Rafael C. Gonzalez. (1975). *Digital image processing. Proc. IEEE* (3rd ed., Vol. 63). <https://doi.org/10.1049/ep.1978.0474>
- Rajbhoj, P. S. M., Gholap, R. A., Dubale, N. N., & Gogawale, A. A. (2016). Real-Time Fire Detection for Video Surveillance Using Digital Image Processing, 3(February 2014), 2014–2017. <https://doi.org/10.15662/IJAREEIE.2016.0503063>
- Rossi, L., Akhloufi, M., & Tison, Y. (2011). On the use of stereovision to develop a novel instrumentation system to extract geometric fire fronts characteristics. *Fire Safety Journal*, 46(1–2), 9–20. <https://doi.org/10.1016/j.firesaf.2010.03.001>
- S.E.Memane, V. S. K. (2015). A Review on Flame and Smoke Detection Techniques in Video ' s, 855–859.
- Santana, P., & Gomes, P. (2012). A vision-based system for early fire detection, 739–744.
- Shi, L., Long, F., Zhan, Y., & Lin, C. (2016). Video-based Fire Detection with Spatio-temporal SURF and Color Features, 258–262.
- Shrivastava, M., & Matlani, P. (2016). A SMOKE DETECTION ALGORITHM BASED ON K-MEANS SEGMENTATION Head Of Department (CSE) Guru Ghasidas University , koni , Bilaspur (C . G) , India Ph . D Scholar , Guru Ghasidas University , koni , Bilaspur (C . G) , India.
- Sounthiraraj, D., Sahs, J., Greenwood, G., Lin, Z., & Khan, L. (2014). SMV-HUNTER: Large Scale, Automated Detection of SSL/TLS Man-in-the-Middle Vulnerabilities in Android Apps. *Symposium on Network and Distributed System Security (NDSS)*, (February), 23–26. <https://doi.org/10.14722/ndss.2014.23205>
- Szeliski, R. (2010). *Computer Vision : Algorithms and Applications*.
- Töreyin, B. U., Dedeoğlu, Y., Güdükbay, U., & Çetin, A. E. (2006). Computer vision based method for real-time fire and flame detection. *Pattern Recognition Letters*, 27(1), 49–58. <https://doi.org/10.1016/j.patrec.2005.06.015>
- Verstockt, S. (2011). Multi-modal video analysis for early fire detection.
- Vinicius, P., Borges, K., Izquierdo, E., & Member, S. (2010). A Probabilistic Approach for

- Vision-Based Fire Detection in Videos, *20*(5), 721–731.
- Wang, H., Yang, W., Jiang, H., & Wang, Y. (2008). Application of Distributed GA-Based RBF Neural Network in Fire Detection. *2008 Fourth International Conference on Natural Computation*, 398–402. <https://doi.org/10.1109/ICNC.2008.851>
- Wang, J., & Wang, H. (2014). Tunable fiber laser based photoacoustic gas sensor for early fire detection. *Infrared Physics & Technology*, *65*, 1–4. <https://doi.org/10.1016/j.infrared.2014.03.002>
- Wang, X., Lo, S., & Zhang, H. (2013). Influence of Feature Extraction Duration and Step Size on ANN based Multisensor Fire Detection Performance. *Procedia Engineering*, *52*, 413–421. <https://doi.org/10.1016/j.proeng.2013.02.162>
- Wang, Y., Yu, C., Tu, R., & Zhang, Y. (2011). Fire detection model in Tibet based on grey-fuzzy neural network algorithm. *Expert Systems with Applications*, *38*(8), 9580–9586. <https://doi.org/10.1016/j.eswa.2011.01.163>
- Weihua, X., Lei, X., Junfeng, L. I., & Xinlong, Z. (2011). Moving Object Detection Algorithm Based on Background Subtraction and Frame Differencing, *5*(5), 3273–3276. <https://doi.org/10.17148/IJARCC.2016.55200>
- Wong, A. K. K., & Fong, N. K. (2014). Experimental Study of Video Fire Detection and its Applications. *Procedia Engineering*, *71*, 316–327. <https://doi.org/10.1016/j.proeng.2014.04.046>
- Wong, T. (2015). Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. *Pattern Recognition*, *48*(9), 2839–2846. <https://doi.org/10.1016/j.patcog.2015.03.009>
- Yadav, G., Gupta, V., Gaur, V., & Bhattacharya, M. (2012). Optimized Flame Detection Using Image Processing Based Techniques. *Indian Journal of Computer Science and Engineering*, *3*(2), 202–211.
- Yu, C., Mei, Z., & Zhang, X. (2013). A Real-time Video Fire Flame and Smoke Detection Algorithm. *Procedia Engineering*, *62*, 891–898. <https://doi.org/10.1016/j.proeng.2013.08.140>
- Yuan, F. (2008). A fast accumulative motion orientation model based on integral image for video smoke detection. *Pattern Recognition Letters*, *29*(7), 925–932. <https://doi.org/10.1016/j.patrec.2008.01.013>
- Yuan, F. (2011). Video-based smoke detection with histogram sequence of LBP and LBPV pyramids. *Fire Safety Journal*, *46*(3), 132–139. <https://doi.org/10.1016/j.firesaf.2011.01.001>
- Zhang, D., & Wang, Y. (2016). Real-Time Fire Detection Using Video Sequence Data. *School of Electrical and Information Engineering*, 3620–3623.
- Zhang, H., Zhang, N., & Xiao, N. (2015). Fire detection and identification method based on visual attention mechanism. *Optik - International Journal for Light and Electron Optics*, *126*(24), 5011–5018. <https://doi.org/10.1016/j.ijleo.2015.09.167>
- Zhang, Y., Tian, W., & Liu, S. (2011). Fire Time Series Forecasting Based on Markov-

- SVR Model. *2011 International Conference on Multimedia and Signal Processing*, (4), 278–281. <https://doi.org/10.1109/CMSP.2011.145>
- Zhao, Y., Li, Q., & Gu, Z. (2015). Early smoke detection of forest fire video using CS Adaboost algorithm. *Optik*, 126(19), 2121–2124. <https://doi.org/10.1016/j.ijleo.2015.05.082>
- Zheng, X., Yuan, Y., & Lu, X. (2016). A Target Detection Method for Hyperspectral Image Based on Mixture Noise Model. *Neurocomputing*, 216, 331–341. <https://doi.org/10.1016/j.neucom.2016.08.015>