



CUARTO CONGRESO INTERNACIONAL
CIENTÍFICO Y TECNOLÓGICO
DE LA PROVINCIA DE BUENOS AIRES

ANÁLISIS DE VIDEO Y CLASIFICACIÓN DE OBJETOS EN UNA PLATAFORMA DE VIGILANCIA DE CÓDIGO ABIERTO UTILIZANDO PROCESAMIENTO DIFERIDO

Laboratorio de Plasmas Densos (PLADEMA)

ANÁLISIS DE VIDEO Y CLASIFICACIÓN DE OBJETOS EN UNA PLATAFORMA DE VIGILANCIA DE CÓDIGO ABIERTO UTILIZANDO PROCESAMIENTO DIFERIDO

A. Pérez, L. Domínguez, J. P. D'Amato, A. Rubiales, L. Fernández Esteberena y R. Barbuzza*

Laboratorio de Plasmas Densos (PLADEMA)
pladema@exa.unicen.edu.ar

RESUMEN

La inseguridad es un problema que afecta a todas las ciudades del mundo. Las más informatizadas recurren a la videovigilancia para combatirla, montando centros de monitoreo con cientos de cámaras y personal idóneo que realiza las tareas de observación. Sin embargo, este método parece no ser suficiente y los organismos públicos deben atender un reclamo social por mayor transparencia y eficiencia en el accionar ante un delito. Aunque existen soluciones corporativas, estas tienden a centralizar el desarrollo y no es sencillo añadir nuevas funcionalidades. En este contexto es que surge el presente proyecto: una plataforma de administración de cámaras y sensores basada en bibliotecas de código abierto, para apoyar la gestión integral de la seguridad. Esta plataforma complementa técnicas básicas de análisis automatizado de videos, tal como la detección de movimiento y seguimiento de objetos, hasta otras específicas, como la clasificación de objetos o la identificación de patentes. La arquitectura desarrollada es abierta, escalable y permite añadir diferentes tipos de algoritmos a partir de un pipeline de análisis diferido. En este trabajo se presenta la estructura ideada, los algoritmos actualmente utilizados para el seguimiento y clasificación de objetos, y diferentes resultados de análisis y eficiencia de la propuesta.

Palabras clave: videovigilancia, plataforma, clasificación de objetos.

* A. Pérez, CICIPBA, PLADEMA-UNCPBA, aperez@exa.unicen.edu.ar; L. Domínguez: PLADEMA-UNCPBA, CONICET, ldominguez@exa.unicen.edu.ar; J. P. D'Amato: PLADEMA-UNCPBA, CONICET, jpdamato@exa.unicen.edu.ar; A. Rubiales: CICIPBA, PLADEMA-UNCPBA, arubiales@exa.unicen.edu.ar; L. Fernández Esteberena: PLADEMA-UNCPBA, CONICET, la.ferest@gmail.com; R. Barbuzza: CICIPBA, PLADEMA-UNCPBA, rbarbu@exa.unicen.edu.ar

INTRODUCCIÓN

La seguridad se ha convertido en uno de los intereses principales de las políticas de gobierno. La preocupación que generan en la sociedad los altos índices de violencia e inseguridad actuales (Kessler, 2009) motiva el estudio de nuevas tecnologías que permitan a los habitantes vivir con mayor tranquilidad. Este hecho se intensifica en grandes ciudades, donde se producen situaciones lamentables con mayor frecuencia. Según Dammert *et al.* (2010), entre el año 2000 y 2008, la tasa de población penitenciaria en América Latina creció un 42 %, y según las tendencias en los buscadores web, en los últimos diez años el interés por términos como “robo”, “hurto” o “inseguridad” se encuentran en constante aumento (Google Trends, 29 de abril de 2017).

Algunos países ya cuentan con casos de éxito para atacar el problema. Por ejemplo, en Londres (Big Brother Watch, 2012), se estima que un individuo puede ser registrado en un día normal por aproximadamente 300 cámaras. Con un propósito similar, Calavia *et al.* (2012) iniciaron un proyecto a partir del cual una extensa red de sensores y dispositivos inalámbricos pretende controlar grandes áreas metropolitanas de forma inteligente.

En Latinoamérica, y en particular en Argentina, los municipios juegan un papel principal y activo en pos de lidiar con la inseguridad (Carrión *et al.*, 2009). En general, para atacar este problema, se ha concebido el desarrollo de grandes centros de monitoreo, basados en tecnología de circuitos cerrados de cámaras. Este modelo comprende la instalación de una red compleja de videocámaras distribuidas en toda la ciudad, con una conectividad exclusiva y con un *software* de gestión y almacenamiento de los videos (Domínguez *et al.*, 2016). Estos videos se almacenan por períodos prolongados para su futura visualización o para ser utilizados como evidencia judicial. La tarea de monitoreo la realizan grupos de personas que observan y reportan sucesos tales como delitos o accidentes, de manera constante (24 horas al día), en horarios rotativos. En caso de un suceso, se recurre a los cuerpos policiales, ya sean del gobierno municipal o provincial, los cuales realizan el control correspondiente y actúan en conjunto con estos centros en la erradicación del delito.

Aunque existen soluciones comerciales que resuelven gran cantidad de los problemas, no siempre se adaptan correctamente a la legislación e intereses de cada país. Incluso, el modelo de “paquete cerrado” que presentan atenta contra el nuevo modelo de gobierno digital, que pretende ser abierto y transparente, tal como lo plantean Ixmatlahua *et al.* (2015).

Asimismo, estos sistemas generalmente siguen un esquema centralizado, donde todo el análisis se realiza en una central de monitoreo que requiere un tipo de equipamiento muy específico. Pensando en ciudades digitales y teniendo en cuenta el bajo costo de un procesador embebido, es natural que se considere distribuir el análisis de imágenes, a fin de escalar el sistema más rápidamente. En este esquema, la solución debe coexistir con múltiples dispositivos conectados que realizan análisis en el lugar, y que reportan únicamente cuando se detecta una situación anormal. Además, para una plataforma accesible, transparente e interconectada, es vital contar con una arquitectura que permita adaptarse a múltiples

situaciones de manera sencilla y que, bajo ciertas condiciones, posibilite el acceso a los datos a agentes y organismos externos al centro de monitoreo.

PROPUESTA DE TRABAJO

Entendiendo que la seguridad es un tema de interés, no solo en el ámbito social sino también en el académico, es que ha surgido el presente trabajo. Nuestro proyecto plantea una plataforma distribuida que permite administrar y compartir, a través de una solicitud web, un conjunto de cámaras IP con procesamiento embebido, en forma pública o privada. Esta idea surge como respuesta a la creciente demanda social en materia de seguridad y en sintonía con el aumento constante en la instalación de cámaras de seguridad que apuntan a la vía pública. A ello se suma una filosofía colaborativa que facilite el cumplimiento de una meta general: mayor seguridad social.

Esta plataforma contempla la automatización de tareas para la detección y seguimiento de personas y objetos en diferentes cámaras de video, de manera integral. Lo innovador de la plataforma es que se encuentra preparada para trabajar con múltiples técnicas de análisis automático que puedan adecuarse a cada situación particular y que incluso puedan ejecutarse en diferentes tipos de dispositivos (servidores, micro-PC, embebidos). Su arquitectura abierta y distribuida contempla el uso de cámaras con procesamiento *in situ* para detectar rápidamente la información de interés en el mismo lugar en que sucede. Solo aquel evento importante será transmitido a una central, lo que permite escalar el sistema para soportar muchas cámaras sin afectar el rendimiento global.

Para extender y parametrizar el análisis automático de acuerdo a diferentes situaciones se propone un mecanismo de análisis en dos etapas. Por un lado, un algoritmo de sustracción de fondo que se ejecuta en tiempo real y discrimina rápidamente objetos en movimiento del escenario estático. Por otro, un análisis diferido que a partir de los objetos detectados extrae información detallada de acuerdo al tipo de objeto.

El trabajo se divide en diferentes secciones. Primero se presentan las herramientas existentes con propósitos similares y un resumen del estado de arte. Luego se describe la arquitectura desarrollada y los detectores utilizados. A continuación se formulan algunos resultados preliminares de los recorridos de una persona, junto con las problemáticas encontradas, y por último, se exponen las conclusiones y los trabajos que quedan pendientes para un futuro.

ESTADO DEL ARTE

Actualmente existen una gran cantidad de sistemas de videovigilancia y, en lo que respecta al seguimiento de personas, hay mucha bibliografía reciente pues se trata de un tema muy estudiado gracias al avance del poder de cómputo de los procesadores actuales. Desde el

punto de vista algorítmico, en trabajos como el de Serby *et al.* (2004), se detallan algunos de los métodos más comunes para la detección y seguimiento de objetos. Hall *et al.* (2010) realizan una comparación interesante entre los distintos detectores, en donde se concluye que la combinación de los mismos puede ser muy útil para disminuir la tasa de falsos positivos sin comprometer el análisis en tiempo real. Más recientemente, en Ojha y Sakhare (2015) se revisan las estrategias generales para la clasificación y el seguimiento de personas en video y en Kim *et al.* (2011) se plantea un método robusto en tiempo real, apto para el seguimiento de objetos a partir de predecir su velocidad y dirección. Además, en Olivieri *et al.* (2012) se describe un método denominado *Motion Vector Flow Instance* (MVFI), capaz de detectar actividades humanas complejas como “caminar” o “correr” con cámaras estáticas.

Considerando las soluciones integrales existentes, en Gdanks (2011) se hace una revisión de 18 sistemas de videovigilancia comerciales y un resumen de los algoritmos más utilizados en etapas de detección, seguimiento y clasificación de distintos objetos.

En otro orden, debido al volumen de información que estos sistemas tienen que procesar, se discuten diferentes arquitecturas que abordan la problemática. En Bramberger *et al.* (2006) se introduce un esquema distribuido, basado en una red de procesadores con cámaras, que permite descentralizar el trabajo pero se ve limitado en el tipo de cámaras soportadas. Si bien en trabajos como los de Lin *et al.* (2011) y Hassan *et al.* (2015) se utiliza la nube para resolver la escalabilidad de la plataforma, los mismos están orientados únicamente al procesamiento de video, descartando la integración de otros tipos de señales.

Procesamiento de imágenes en la nube

El procesamiento en la nube se ha vuelto un modelo viable de servicios distribuidos a través de Internet, gracias a las facilidades de virtualización (Mahjoub *et al.*, 2011). Dependiendo de los recursos físicos disponibles en los nodos de la nube, se pueden ejecutar varias máquinas virtuales con diferente funcionalidad, capacidades de procesamiento y datos. Migrar aplicaciones de análisis de imágenes y datos a la nube puede permitir a algunas organizaciones reducir costos operativos, al mismo tiempo que se incrementa la escalabilidad, la eficiencia y la robustez. Existen proyectos relacionados en el área de imágenes médicas, tal como los de Bastiao Silva *et al.* (2011) y Ahmed y Abdullah (2011), que utilizan servicios en la nube para la gestión.

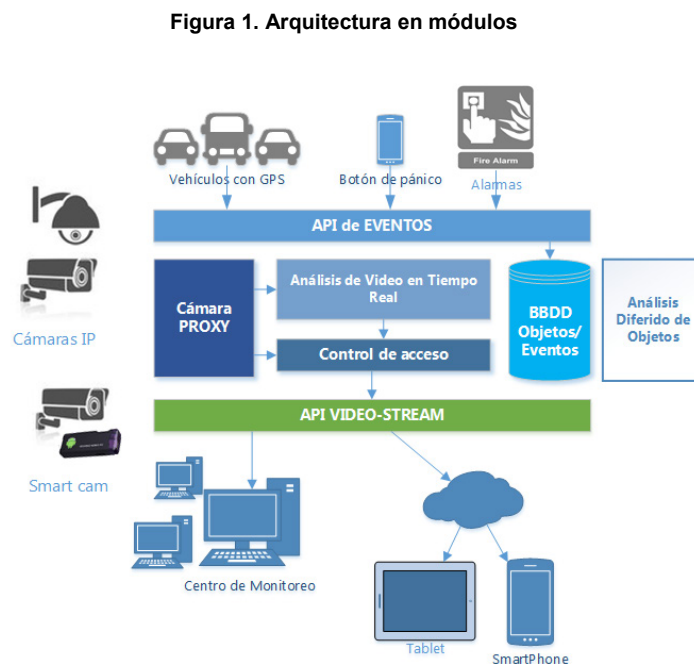
Otras alternativas de servicios trabajan con análisis de imágenes en general. Algunas iniciativas son llevadas por universidades, como el proyecto CloudCV (Agrawal *et al.*, 2015) y otras ligadas a empresas, como Watson¹ o Digits². Muchas ofrecen caracterización de imágenes usando redes muy entrenadas, pero no son gratuitas y tienen limitaciones en el acceso. De todas formas, son alternativas interesantes para poder incorporar en un sistema como el propuesto, siempre que el tiempo de respuesta y el costo no sea muy elevado.

¹ Véase IBM. *Watson Developer Cloud*, <<https://www.ibm.com/watson/developercloud/>>. Último acceso: 29 de abril de 2017.

² Véase NVIDIA, *Digits - Interactive Deep Learning GPU Training System*, <<https://developer.nvidia.com/digits>>. Último acceso: 29 de abril de 2017.

PLATAFORMA PROPUESTA

La plataforma contempla la integración de cámaras IP, junto con herramientas de análisis automático y un gestor de eventos georreferenciados, preservando y controlando el acceso a los datos. Esta funcionalidad se agrupa por capas; cada una contempla diferentes módulos y tecnologías distribuidas. En la figura 1, se muestran los diferentes componentes de la arquitectura. Esta arquitectura ya fue propuesta en D'amato *et al.*, (2016) y ahora se amplía para soportar un nuevo esquema de procesamiento.



El principal objetivo es detectar comportamientos inusuales de manera automática y alertar al operador del centro de monitoreo para que pueda tomar una acción. Este proceso se puede realizar mediante técnicas automáticas de caracterización de objetos o de situaciones, de manera de identificar eventos que requieran la intervención humana.

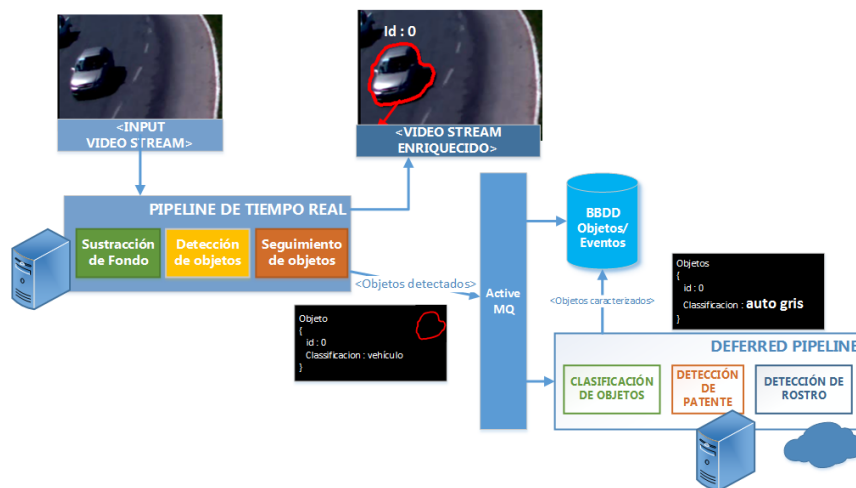
A todas las cámaras se accede a través de un servidor *proxy* que hace de intermediario en las peticiones de recursos que realiza un cliente a un servidor. Esta situación estratégica de punto intermedio le permite ofrecer diversas funcionalidades: control de acceso, registro del tráfico, restricción a determinados tipos de tráfico, mejora de rendimiento, anonimato de la comunicación, caché web, etc. Las cámaras generan nuevos *streams* de video que ingresan a la plataforma a través de este servidor.

Los algoritmos de clasificación y caracterización de objetos pueden ser de múltiples tipos y con diferentes objetivos. Desde algoritmos más simples de clasificación por la forma y tamaño, hasta otros específicos como detectores de patentes y rostros. Cada algoritmo tiene una complejidad computacional diferente y utilizan diferentes características de las imágenes observadas. En cualquiera de los casos, lo que se pretende es poder describir un objeto con el mayor detalle posible.

Para este fin, se propone en este trabajo analizar cada nueva imagen que proviene desde una cámara, de diferentes maneras. Por un lado, aplicar un algoritmo de detección de objetos en una zona de interés, discerniendo entre el fondo estático y aquellos objetos en movimiento, realizado de manera continua por cada nueva imagen capturada. Esto constituye el *pipeline* de tiempo real. Una de las salidas de este proceso es un *stream* enriquecido que indica los objetos detectados.

Luego, para cada objeto detectado, es importante extraer otras características más detalladas y de mayor complejidad, necesarias para tareas como la clasificación de objetos o la identificación de patentes. Dado que este análisis es en general más costoso en cuanto a lo computacional, se realiza en modo asincrónico (denominado también *pipeline* diferido). En la figura 2 se presenta un esquema del funcionamiento.

Figura 2. Esquema de funcionamiento



Finalmente, los objetos detectados se almacenan en una base de datos distribuida, que es accesible por diferentes dispositivos de la red. A continuación, se describe cómo se comunican los módulos de procesamiento y los algoritmos de clasificación utilizados.

Detección del movimiento en tiempo real y distribución del procesamiento

El primer paso para la detección de eventos es poder determinar si hubo o no movimiento a partir de una serie de imágenes. Este problema es análogo al problema de sustracción de fondo, que calcula la variación entre imágenes consecutivas (Shaikh *et al.* 2014 y Piccardi, 2004). Los métodos varían en complejidad computacional y eficacia del resultado. Actualmente, en la plataforma se utilizan variantes de VIBE (Bouwmans *et al.*, 2014), ya que este algoritmo permite descartar movimiento continuo, como por ejemplo el movimiento de las hojas de los árboles.

Cuando las cámaras son exteriores, se suceden otros factores que además varían dependiendo del momento del día, la nubosidad y la época del año. Para contrarrestar estos

efectos, se suelen utilizar ventanas de movimiento, como se menciona en el sitio web de ActiveMQ³. Este esquema permite aplicar una operación lógica para procesar únicamente los movimientos que se detecten dentro de las dimensiones de las ventanas de movimiento y descartar todos aquellos que se produzcan por fuera.

Estos algoritmos tienen un alto costo computacional, por lo que se requiere un procesador dedicado a esta tarea. Para mejorar la eficiencia del sistema general, se concibió la idea de llevar el procesamiento a la ubicación de la videocámara. Para esto, se diseñaron gabinetes estancos con la capacidad de alojar PC industriales de tamaño pequeño conectadas directamente a la cámara. Estas PC realizan el análisis de los objetos y transmiten a la central el resultado obtenido, junto con la imagen del video.

En general, la salida de este paso es un conjunto de píxeles (agrupados o no) que han sido marcados como representativos de objetos en movimiento. En caso de que se detecte un movimiento en cada zona, se determinan sus características de movimiento tales como el tamaño del objeto, la dirección de movimiento y las últimas posiciones donde ha aparecido aplicando técnicas de seguimiento. Si los objetos detectados tienen un tamaño pequeño, son descartados. En otro caso, se los identifica de manera unívoca dentro de la red, y se informa al sistema a fin de registrarlo y caracterizarlo.

Comunicación entre módulos

Los módulos de análisis de imágenes pueden ejecutar en servidores de procesamiento o incluso en computadoras embebidas conectadas directamente a la cámara, y que alerten solo ante un evento de interés. La comunicación entre los diferentes módulos de la arquitectura se basa en un esquema de pasaje de mensajes. La tecnología utilizada para implementar este esquema se llama ActiveMQ⁴. Cada nuevo tipo de evento se publica en la red como un mensaje en un “tópico”. Cada módulo puede ser productor o consumidor de este “tópico”. Entre estos módulos también se encuentra el encargado de registrar en la base de datos y los módulos de visualización.

En el caso de seguimiento de objetos, a partir de una detección de movimiento en el *pipeline* de tiempo real, se genera un mensaje determinado de *objeto nuevo*. Este mensaje incluye el ID del objeto, un recorte de la imagen asociada y, si es necesario, otra información que pueda inferirse (por ejemplo, la dirección y velocidad de desplazamiento del objeto). El mensaje y las imágenes se serializan en formato JSON y se publican en la plataforma. Aquellos módulos que escuchen ese tópico toman el mensaje, lo deserializan, lo procesan y lo publican nuevamente con otro tópico para su registro.

³ Véase ActiveMQ, <<http://activemq.apache.org/>>. Último acceso: 29 de abril de 2017.

⁴ Véase The Apache Software Foundation, <<http://activemq.apache.org/>>. Último acceso: 29 de abril de 2017.

Algoritmos de pipeline diferido

Existen múltiples algoritmos especialmente desarrollados para la detección de personas y vehículos. Muchos de estos algoritmos son de código abierto y otros a través del uso de librerías de terceros. Los algoritmos varían en velocidad, precisión, y tipo de datos necesarios. Algunos utilizan una imagen del video, otros una secuencia de imágenes, o en algunos casos sólo alguna información como el movimiento realizado. El resultado es una caracterización más detallada del objeto, que permite una mejor clasificación.

En el ámbito de la seguridad urbana, se detectó que es muy importante poder distinguir con una alta precisión el *tipo* del objeto, el *color* principal o el *tamaño*. Para el caso de los vehículos, también es importante poder discriminar su *patente*. En cuanto a las personas, es importante determinar su *comportamiento* a partir del movimiento. Al contar con esta información, es posible registrar y rastrear rápidamente al objeto de interés dentro de una gran *base de datos*. El proceso de caracterización de objetos puede realizarse en un algoritmo ejecutándose en un cliente de la red (puede ser el mismo que captura el video), o en algoritmos remotos alojados en la nube.

Cliente de procesamiento local

Para discriminar el tipo de objetos, se utilizaron algunos clasificadores de baja complejidad. Estos algoritmos se basan en atributos rápidamente identificables, tal como la distribución del color o la forma de un objeto. En efecto, sus objetivos son:

- La clasificación de los objetos detectados por tamaño y por deformación.
- La detección del color del objeto.

El esquema de estos algoritmos se muestra en la figura 3.

Figura 3. (Izq.) Análisis del objeto por deformación. (Der.) Análisis del objeto por color



Algoritmos basados en clasificadores clásicos, como AdaBoost o SVM, no son siempre aplicables, sobre todo porque cada cámara tiene un punto de observación distinto y requiere reentrenar los algoritmos con un costo computacional muy elevado.

Cliente para el procesamiento en la nube

Google, Microsoft e IBM han publicado sus servicios de análisis de imágenes a partir de API para consultar el tipo de imagen observada. Dichas API se ejecutan con redes de clasificadores basados en DNN muy entrenados, que se pueden utilizar para el fin que aquí se busca. Estos servicios requieren una imagen con buen detalle.

Para ejecutar estas técnicas en conjunto con la arquitectura, se requiere un cliente en la red que reciba los mensajes de nuevos objetos. Ante cada mensaje, se genera la invocación a la API en la Nube, se espera el resultado obtenido y luego se actualiza la base de datos.

Análisis en paralelo y resolución de múltiples soluciones

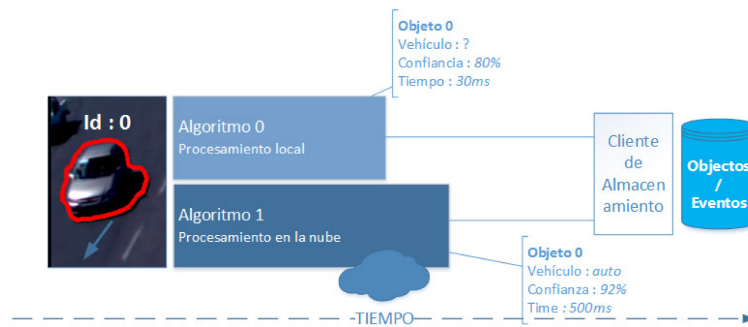
Por cada nuevo objeto detectado en tiempo real, se genera un mensaje asociado a un tópico. A este se le asocia una imagen que lo representa y el tiempo de captura, en caso de que se deba acceder al *stream* de video para ubicarlo. Este tópico se publica a través de la red, por lo cual pueden existir múltiples clientes de procesamiento que reciban el mismo tópico. Dada la versatilidad de la arquitectura, se pueden iniciar múltiples clientes que ejecuten diferentes algoritmos para analizar el mismo objeto.

Una vez que los algoritmos reciben y computan la solución, esta es publicada nuevamente en la red para ser almacenada en la *base de objetos*. La salida de estos algoritmos también sigue en formato JSON, pero cada uno tiene diferentes tiempos de respuesta, por su complejidad y por la latencia de la red. En algunos casos (por ejemplo, una alarma por intrusión), si el tiempo de demora es muy alto, el resultado pierde sentido; en otros, es indispensable tener mayor precisión (por ejemplo, en una detección de patente).

Si para un caso existen múltiples soluciones simultáneas, se pondera cuál es la solución más conveniente. Esta ponderación se realiza en el momento previo al almacenamiento. Cada mensaje de respuesta viene asociado con información del algoritmo que lo realizó, tal como el tipo de análisis, el tiempo de respuesta y la confiabilidad del resultado.

De acuerdo al tipo de análisis, se aplican dos reglas de almacenamiento: “primera respuesta recibida” o “respuesta más confiable”. Esta información se asigna al momento de instanciar la arquitectura, como se muestra en la figura 4.

Figura 4. Esquema de dos algoritmos distribuidos para caracterizar un objeto



RESULTADOS

La plataforma propuesta se encuentra en funcionamiento a nivel de prototipo y en constante desarrollo por parte del presente grupo. Se ha instalado en un campus universitario ubicado en la ciudad de Tandil, Buenos Aires. Estas cámaras se conectan a través de una red interna de la universidad, con una velocidad promedio de 100 mbps.

Todos los *streams* de los videos se registran y gestionan a través de una versión de escritorio de la plataforma, que puede ser administrada en su totalidad. La base de datos utilizada es SQL Server, con una licencia para educación, aunque puede funcionar en otras de licencia libre. A su vez, a través de un portal web, implementado en el *framework* AngularJS, se permite acceder a estas cámaras, y así detectar las zonas observadas. El módulo web puede también ser accesible a través de móviles y *tablets*, con una interfaz adaptable, incluso en redes de baja velocidad.

Para implementar la comunicación entre módulos, se utiliza ActiveMQ, el cual cuenta con múltiples implementaciones en diferentes lenguajes de programación y sistemas operativos (Windows, Linux, Android, entre otros). Esto permite conectar aplicaciones desarrolladas en diferentes tecnologías.

Con el fin de analizar la factibilidad de la distribución del procesamiento, se realizaron diversas pruebas con la arquitectura propuesta, con los diferentes clasificadores desarrollados. Para estas pruebas, se configuró un servidor, un *host* de procesamiento local y un *host* consultando un servicio en la nube (Agrawal *et al.*, 2015), con las siguientes alternativas:

- Configuración 1: un único equipo con capacidades de detección y análisis.
- Configuración 2: equipo *ad hoc* para detección de objetos – Servidor para análisis.

Tal como se nombró en la arquitectura desarrollada, se utiliza el algoritmo VIBE para detectar objetos en movimiento y una biblioteca propia para el seguimiento de objetos dentro del *pipeline* de tiempo real. Se utilizaron cámaras de alta resolución conectadas directamente a una computadora local y un servidor en una máquina virtual para almacenar todos los objetos detectados.

Sobrecarga de la distribución del procesamiento

En el primer tipo de prueba, se buscó evaluar el impacto de distribuir el procesamiento en diferentes procesadores de acuerdo a la complejidad del algoritmo y al tamaño de la imagen. Se probó con las dos configuraciones, con imágenes de objetos de baja resolución (video de 640x480) y de alta resolución (video de 1920x1280).

De la imagen, se transmite únicamente el recorte donde se encuentra el objeto detectado, en formato comprimido. Se midió el tiempo que añade serializar y transmitir por ActiveMQ esta detección. En la tabla 1 se muestran los resultados obtenidos, aplicados a 1000 imágenes.

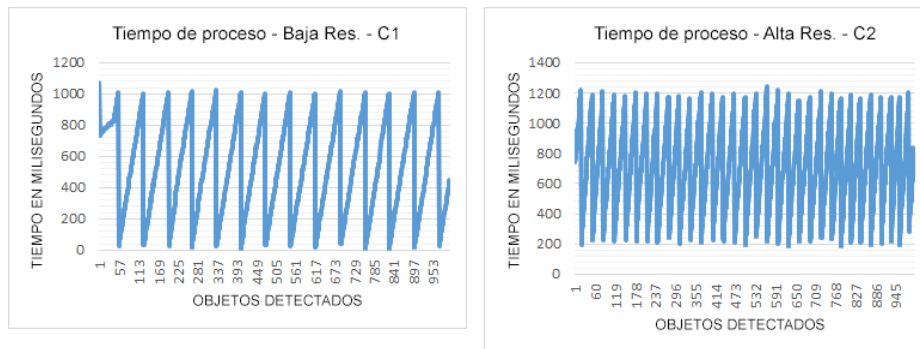
Tabla 1. Inferencia del tiempo de transferencia con respecto al procesamiento

		Tiempo transferencia	Tiempo procesamiento	Sobrecarga (%)
Configuración 1	Low Res	537,4	15,6	97%
	High Res	879,8	327,6	62%
Configuración 2	Low Res	494,8	9,5	97%
	High Res	708,6	170,4	75%

Como puede observarse en la tabla 1, la sobrecarga de transferencia es considerablemente alto, aun más al operar con imágenes de baja resolución. También, se observó que al procesar en el servidor (configuración 2), se reduce el tiempo de procesamiento en un 50 % aproximadamente. Al tratarse de algoritmos simples, el tiempo de procesamiento es relativamente bajo. Al utilizar técnicas más sofisticadas, los tiempos de transferencia se mantienen constantes. De igual forma se obtiene el tiempo total (procesamiento + transferencia) necesario para cada imagen (figura 5).

Lo que se observó es que las imágenes son encoladas, en el servidor o en el cliente, aumentando los tiempos de procesamiento. Este comportamiento es observado en las cuatro configuraciones. El problema surge cuando las imágenes son serializadas/deserializadas. Durante este paso, la transmisión es bloqueada y los mensajes son encolados.

Figura 5. Tiempo de caracterización de objetos para las configuraciones 1 y 2



En la figura 5 puede verse que el proceso completo de clasificación de una imagen toma a lo sumo 1,2 segundos. En contraposición con un sistema industrial de tiempo real, donde un segundo es crítico, en un sistema de vigilancia, este tiempo es insignificante comparado con el tiempo que puede tomarle a una patrulla o a una ambulancia arribar al lugar de un incidente. Sin embargo, el uso del *pipeline* diferido permite al sistema tener gran flexibilidad en el uso de diferentes métodos de clasificación y procesamiento. Incluso varios algoritmos pueden ser ejecutados en paralelo. Adicionalmente, no es necesario que todos los algoritmos sean implementados dentro de la plataforma, ya que pueden utilizarse servicios de terceros e incluso ser procesados en la nube.

Comparación del mismo análisis con diferentes métodos

Una segunda serie de pruebas se llevaron a cabo con diferentes estrategias para caracterizar objetos por color y tipo. Heurísticas desarrolladas basadas en formas y conteo de píxeles de color fueron comparadas con el servicio de análisis provisto por IBM (Watson Visual Recognition) en la nube. Los servicios de IBM construyen una lista de las posibles clases que coinciden con imágenes previamente clasificadas, incluyendo la probabilidad de coincidencia. En este trabajo, se considera la clase más probable como clase elegible. Esta consideración se aplica también a la determinación del color, en la cual se producen clasificaciones incorrectas. Sesenta objetos fueron clasificados y los resultados se muestran en la tabla 2.

A primera vista, Watson es más fiable al describir tipos de objetos en comparación con nuestro método. La precisión de color es menor en este momento. Al realizar diferentes pruebas, el reconocimiento de color en Watson mejora al utilizar imágenes sin recortes, por lo cual se propone evaluar nuevamente las imágenes.

Tabla 2. Servicios en la nube y comparación de algoritmos heurísticos

	Precisión clasificación por tipo	Precisión detección de color	Tiempo de transferencia y procesamiento
Heurística	73.3 %	65.0 %	1,2 s
Watson	93.3 %	13.3 %	36,8 s

El tiempo de transferencia y procesamiento es altamente influenciado por el tiempo de proceso del servicio.

CONCLUSIONES

En este trabajo se presentó una plataforma distribuida para la gestión de múltiples cámaras y sensores, con el soporte para la gestión integral de eventos. La plataforma contempla la definición de una API para integrar aplicaciones externas y la manipulación de cámaras IP propias de un organismo o externas, aplicando un control de los accesos a través de un servidor *proxy*. Principalmente se discutió una variación de la plataforma de manera de poder aplicar algoritmos diferentes para analizar objetos. Para esto, se han desarrollado y aplicado algoritmos de caracterización, a partir de un esquema diferido.

El esquema propuesto puede ser fácilmente adaptado para incluir nuevos algoritmos bajo demanda. Si bien los resultados indican que el paso de transmisión reduce el desempeño general del sistema, debe considerarse que los procesos involucrados, cuando se activa una alarma de un sistema de seguridad, disparan acciones que pueden tardar minutos. Este hecho da lugar al uso del *pipeline* diferido y a servicios en la nube, tomando ventaja de su flexibilidad y procesamiento paralelo.

Como fue demostrado, un buen resultado puede obtenerse al combinar ambas estrategias. La configuración final de la plataforma depende de qué se priorice: o bien el tiempo de respuesta o bien la confianza en los resultados. En un trabajo futuro, se pretende perfeccionar las técnicas de análisis y la incorporación de nuevos tipos de dispositivos que funcionen como disparadores de eventos. El objetivo es, a su vez, perfeccionar el método de distribución del procesamiento, de manera de generar un balance óptimo de carga de la red y de cada unidad de cálculo.

AGRADECIMIENTOS

Este proyecto se emplaza en la infraestructura de conectividad provista por la Universidad Nacional del Centro de la Provincia de Buenos Aires (UNCPBA). A su vez, ha recibido subsidios de la CICPBA, en la convocatoria Fortalecimiento de Centros CIC 2016.

BIBLIOGRAFÍA

- AGRAWAL, H.; MATHIALAGAN, C. S.; GOYAL, Y.; CHAVALI, N.; BANIK, P.; MOHAPATRA, A. *et al.* (2015). "CloudCV: Large Scale Distributed Computer Vision as a Cloud Service". En HUA, G. Y HUA, X. *Mobile Cloud Visual Media Computing*. Berlín: Springer International Publishing, pp. 265-290.
- AHMED, S. y ABDULLAH, A. (2011). "E-healthcare and Data Management Services in a Cloud". En *8th International Conference on High-capacity Optical Networks and Emerging Technologies*. IEEE, pp. 248-252.
- BASTIAO SILVA, L. A.; COSTA, C.; SILVA, A. y OLIVEIRA J. L. (2011). "A PACS Gateway to the Cloud". En *2011 6th Iberian Conference on Information Systems and Technologies (CISTI 2011)*. IEEE, pp. 1-6.
- BIG BROTHER WATCH (2012). "The Price of Privacy: How Local Authorities Spent Half a Billion Pounds on CCTV in Four Years", A Big Brother Watch report. Recuperado de <<https://bigbrotherwatch.org.uk/2012/02/price-privacy-councils-spend-521m/>>
- BOUWMANS, D.; PORIKLI, F.; HOFERLIN, B. y VACAVENT, A. (2014). *Background Modeling and Foreground Detection for Video Surveillance*. Londres: Chapman and Hall.
- BRAMBERGER, M.; DOBLANDER, A.; MAIER, A.; RINNER, B. y SCHWABACH, H. (2006). "Distributed Embedded Smart Cameras for Surveillance Applications". *Computer*, vol. 39, n.º 2, pp. 68-75.
- CALAVIA, L.; BALADÓN, C.; AGUIAR, J. M.; CARRO, B. y SÁNCHEZ-ESGUEVILLAS, A. (2012). "A Semantic Autonomous Video Surveillance System for Dense Camera Networks in Smart Cities". *Sensors*, vol. 12, n.º 8, pp. 10407-10429.
- CARRIÓN, F.; PONTÓN, C. y ARMIJOS, B. (2009). *120 Estrategias y 36 experiencias de seguridad ciudadana*. Quito: Flacso.
- D'AMATO J. P.; DOMÍNGUEZ, L.; PÉREZ A. y RUBIALES A. (2016). "Plataforma abierta de gestión de cámaras IP y aplicaciones móviles para la seguridad civil ciudadana". *Revista Ibérica de Sistemas e Tecnologías de Inormação*, vol. 20, pp. 48-61.
- DAMMERT, L.; SALAZAR, F.; MONTT, C. y GONZÁLEZ, P. (2010). *Crímen e inseguridad: indicadores para las Américas*. Washington DC: FLACSO-Chile/Banco Interamericano de Desarrollo.
- DOMÍNGUEZ, L.; PÉREZ, A.; RUBIALES, A.; D'AMATO, J. y BARBUZZA, R. (2016). "Herramientas para la detección y seguimiento de personas a partir de cámaras de seguridad". En RedUNCI. *XXII Congreso Argentino de Ciencias de la Computación*. San Luis: Universidad Nacional de San Luis, pp. 251-260.
- GDANKS, KU (2011). "Deliverable 2.1. Review of existing smart video surveillance systems capable of being integrated with ADDPRIV project". Recuperado de <<http://www.addpriv.eu/uploads/public%20deliverables/149-->

GOOGLE TRENDS. Recuperado el 29 de abril de 2017 de <<https://trends.google.com.ar/trends/>>

HALL, D.; NASCIMENTO, J. y RIBEIRO, P. (2005). "Comparison of Target Detection Algorithms Using Adaptive Background Models". En: *2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*. IEEE, pp 113–120. doi: 10.1109/VSPETS.2005.1570905

HASSAN, M. M.; HOSSAIN, M. A.; ABDULLAH-AL-WADUD, M.; AL-MUDAIHESH, T.; ALYAHYA, S. y ALGHAMDI, A. (2015). "A Scalable And Elastic Cloud-Assisted Publish/Subscribe Model for IPTV Video Surveillance System". *Cluster Computing*, vol. 18, n.º 4, pp. 1539-1548.

IBM. *Watson Developer Cloud*. Recuperado el 29 de abril de 2017 de <<https://www.ibm.com/watson/developercloud/>>

IXMATLAHUA, S. D.; RAYGOZA, R. O.; ROMERO, O.; URIBE, F. y VARGAS, E. J. (2015). "Metrópoli digital: una plataforma web para la inclusión integral de las PyMES, sociedad y gobierno en el uso de las tecnologías de la información en la región de las altas montañas del estado de Veracruz, México". *RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação*, n.º 3, pp. 43-54. doi: 10.17013/risti.e3.43-54

KESSLER, G. (2009). *El sentimiento de inseguridad: sociología del temor al delito*. Buenos Aires: Siglo XXI.

KIM, J. S.; YEOM, D. H. y JOO, Y. H. (2011). "Fast and Robust Algorithm of Tracking Multiple Moving Objects for Intelligent Video Surveillance Systems". *IEEE Transactions on Consumer Electronics*, vol. 57, n.º 3, pp. 1165-1170. doi: 10.1109/TCE.2011.6018870

LIN, C. F.; YUAN, S. M.; LEU, M. C. y TSAI, C. T. (2012). "A framework for scalable cloud video recorder system in surveillance environment". En: *Proceedings of the 2012 9th International Conference on Ubiquitous Intelligence and Computing and 9th International Conference on Autonomic and Trusted Computing*. Washington: IEEE, pp. 655-660.

MAHJOUR, M.; MDHAFFAR, A.; HALIMA R. B. y JMAIEL M. (2011). "A Comparative Study of the Current Cloud Computing Technologies and Offers". En *2011 First International Symposium on Network Cloud Computing and Applications*. IEEE, pp. 131-134, 21-23. doi: 10.1109/NCCA.2011.28

NVIDIA. *Digits - Interactive Deep Learning GPU Training System*. Recuperado el 29 de abril de 2017 de <<https://developer.nvidia.com/digits>>

OJHA, S. y SAKHARE, S. (2015). "Image processing techniques for object tracking in video surveillance - A survey". En: *2015 International Conference on Pervasive Computing (ICPC)*, pp. 1-6. doi: 10.1109/PERVASIVE.2015.7087180

OLIVIERI, D. N.; CONDE, I. G. y SOBRINO, X. A. V. (2012). "Eigenspace-based Fall Detection and Activity Recognition from Motion Templates and Machine Learning". *Expert Systems with*

Applications, vol. 39, n.º 5, pp. 5935-5945.

PICCARDI, M. (2004). "Background Subtraction Techniques: a Review". En: *2004 IEEE International Conference on Systems, Man & Cybernetics*, vol. 4. IEEE, pp 3099-3104.

SERBY, D.; KOLLER-MEIER E. y GOOL, L. (2004). "Probabilistic Object Tracking Using Multiple Features". En: KITTLER, J.; PETROU, M. y NIXON, M. S. (Eds.). *Proceedings of the 17th International Conference on Pattern Recognition (ICPR 2004)*, vol. 2, pp. 184-187. doi: 10.1109/ICPR.2004.1334091

SHAIKH, S. H.; SAEED, K. y CHAKI, N. (2014). *Moving Object Detection Using Background Subtraction*. Berlín: Springer International Publishing.

THE APACHE SOFTWARE FOUNDATION. *ActiveMQ*. Recuperado el 29 de abril de 2017 de <<http://activemq.apache.org/>>.