UNIVERSIDAD NACIONAL DEL ALTIPLANO

ESCUELA DE POSGRADO

MAESTRÍA EN INGENIERÍA DE SISTEMAS

TESIS

DETECCIÓN AUTOMÁTICA DE EVENTOS INUSUALES EN IMÁGENES Y VIDEO DE CÁMARAS DE VIGILANCIA

PRESENTADA POR:

ALFREDO CCARI SUCASACA

PARA OPTAR EL GRADO ACADÉMICO DE:

MAGISTER SCIENTIAE EN INGENIERÍA DE SISTEMAS

PUNO, PERÚ

2019

UNIVERSIDAD NACIONAL DEL ALTIPLANO

ESCUELA DE POSGRADO

MAESTRÍA EN INGENIERÍA DE SISTEMAS

TESIS

DETECCIÓN AUTOMÁTICA DE EVENTOS INUSUALES EN IMÁGENES Y VIDEO DE CÁMARAS DE VIGILANCIA

PRESENTADA POR:

ALFREDO CCARI SUCASACA

PARA OPTAR EL GRADO ACADÉMICO DE:

MAGISTER SCIENTIAE EN INGENIERÍA DE SISTEMAS

APROBADA POR EL SIGUIENTE JURADO:

PRESIDENTE

M.Sc. HUGO YOSEF GOMEZ QUISPE

PRIMER MIEMBRO

Mg. ROBERT ANTONIO ROMERO FLORES

SEGUNDO MIEMBRO

D.Sc. DONIA ALIZANDRA RUELAS ACERO

ASESOR DE TESIS

D.Sc. YALMAR PONCE ATENCIO

Puno, 14 de diciembre de 2018

ÁREA: Inteligencia Artificial.
TEMA: Detección Automática De Eventos Inusuales En Imágenes Y Video De Cámaras De Vigilancia.
LÍNEA: Inteligencia Artificial

DEDICATORIA

A Dios, por haberme dado la vida y la fuerza
En quien confió y doy gracias por todo lo
que me ha dado en esta vida.

A mi querido padre Gerardo Ccari Laura y a mi querida madre Isabel Sucasaca Calsin, y a todos mis queridos hermanos Claudia, Víctor, Ruth y a mi querida esposa por todo el apoyo que me brindaron durante toda esta etapa de mis estudios.

AGRADECIMIENTOS

- A Dios por darme la bendición y darme las fuerzas de seguir adelante y estar conmigo en todo momento de mi vida.
- A la Universidad Nacional del Altiplano, por haberme permitido formarme en ella.
- Al Dr. Yalmar T. Ponce Atencio, asesor del trabajo de investigación; por su apoyo y la orientación en el desarrollo de este proyecto de investigación.
- Al jurado calificador de tesis M.sc. Hugo Yosef Gomez Quispe, Mg. Robert Antonio Romero Flores, Dr. Donia Alizandra Ruelas Acero
- A todos mis amigos, Javier Ccalli Olvea, Edward Quisocala Machaca

ÍNDICE GENERAL

Pág.

CAPÍTULO II

PLANTEAMIENTO DEL PROBLEMA

CAPÍTULO III

METODOLOGIA

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

ÍNDICE DE ANEXOS

RESUMEN

La detección automática de eventos inusuales en imágenes y cámaras de video de vigilancia son problemas desafiantes para la visión por computadora, y es un tema de interés especial y de diversos tipos de aplicativos tales como son: la detección de peleas, identificación de asaltos, robos, acciones sospechosas de transeúntes, etc. El interés del desarrollo de este trabajo de investigación es por el avance de las tecnologías y el aumento de las cámaras de vigilancia, por lo cual podemos encontrar en distintos lugares, en cualquier ciudad, del país y el mundo entero. Debido a que hay una inseguridad permanente y se tiene la tecnología necesaria y es por en cuanto se decide desarrollar y utilizar los algoritmos y las técnicas para la detección de eventos y acciones inusuales en humanos y para luego probarlo en videos por medio de las cámaras de vigilancia. Para dicha aplicación se hace uso del algoritmo y técnica de "máquinas de postura convolucional", en la primera fase se hará el uso de los campos de afinidad de las partes del cuerpo, para encontrar los puntos de interés (articulaciones) luego se aplica una red neuronal convolucional y adicionalmente la técnica se apoya en el uso de grafos de correspondencia para poder articular el cuerpo. Empleando los puntos y partes obtenidas por la máquina de postura convolucional se encuentra un esqueleto para cada individuo en la escena, y finalmente comparamos (con una técnica de correspondencia de formas) con posturas predefinidas para poder saber lo que un individuo está haciendo.

Palabras clave: estimación de postura humana, máquinas de soporte de vectorial, máquinas de posturas convolucionales, openpose, Procesamiento de imágenes, redes neuronales convolucionales.

ABSTRACT

Automatic detection of unusual events in images and video surveillance cameras are challenging problems for computes vision, and is a topic of special interest for different types of applications such as: the detection of fights, identification of assaults, robberies, actions passersby suspects, etc. the interest of the development of this research work is for the advancement of technologies and the increase of video surveillance cameras, which we can find in different places, in any city, in the country and throughout the world. Because there in permanent insecurity and it has technology to decide to develop and use the techniques and algorithms for the detection of unusual events and actions in humans and the test it on videos captured by surveillance cameras. For this application, the "convolutional position machines" technique is used, in the first stage the affinity fields of the body parts are used, to find the points of interest (joints) the a red convolutional neural is applied and additionally the technique relies on the use correspondence graphs to articulate the body. Using the points and parts obtained by the convolutional position machine is a skeleton for each individual in the scene, and finally we compare (with a technique of correspondence of forms) with predefined postures to know what an individual is doing.

Keywords: convolutional posture machines, convolutional neural networks, human position methods, image processing, openpose, vector support machines.

INTRODUCCIÓN

La detección automática de eventos inusuales en imágenes y video de cámaras de vigilancia es un trabajo de investigación que ha adquirido un interés especial en estos últimos años, y toda esta tendencia se ha dado debido al incremento de las tecnologías de las cámaras de video vigilancia y producción de videos obtenidos por motivos de seguridad. Dentro de este ámbito de la detección de eventos inusuales son de distintos tipos como, por ejemplo, la detección de robos, peleas en las calles, asaltos de transeúntes, actividades en centros comerciales, etc.

La video vigilancia se ha convertido en una gran necesidad en la vida de las personas, ya que es una manera para poder verificar lo que está ocurriendo y lo que ha ocurrido en un determinado lugar. Así, actualmente, se vienen instalando cámaras de video vigilancia masivamente, tanto en lugares públicos como privados. Todo este despliegue de recursos se da por la necesidad de obtener información en tiempo real y tratar de evitar riesgos que podrían terminar en tragedias. La detección automática de eventos inusuales mediante las cámaras de video vigilancia es un tema de interés y el estado del arte refiere al uso de múltiples técnicas que trabajan en relación a la estimación de postura de una persona y luego poder determinar qué tipo de acción está realizando, en tiempo real. Esto puede permitir interpretar un evento y poder dar una alerta en caso de alguna situación inusual o sospechosa.

En esta investigación se ha usado la técnica para estimación de posturas mediante la aplicación de una red neuronal convolucional, que a su vez usa el método de máquinas de posturas convolucionales y los campos de afinidad de partes.

CAPÍTULO I
REVISIÓN DE LITERATURA

1.1 Marco teórico

1.1.1 Marco conceptual

En este capítulo se efectúa una revisión literaria sobre conceptos de interés general de temas de redes neuronales convoluvionales, máquinas de soporte vectorial, inteligencia artificial y redes neuronales, que serán de gran uso para el desarrollo y la detección de eventos y acciones inusuales en secuencia de video.

1.1.2 Inicialización del modelo

El inicio de la captura de movimiento humano basada en visión computacional, requiere frecuentemente el significado de un modelo humanoide que aproxime la apariencia, forma, estructura cinemática y la postura inicia la del sujeto a ser rastreado. La inicialización requiere el conocimiento previo de lo que constituye un individuo. Tal conocimiento puede ser separado en categorías de, estructura cinemática forma 3D apariencia de color y estimación de partes del cuerpo. (Hernández García, García Reyes, Ramos Cózar, & Guil Mata, 2014)

La mayor parte de los marcos de seguimiento basados en visión adquieren una estructura cinemática humanoide inicial que incorpora un número fijo de articulaciones con grados de flexibilidad especificados. La inicialización cinemática es entonces limitada a la estimación de las longitudes de las extremidades. Los marcos de captura de movimiento basados en marcadores comerciales normalmente obligan a un grupo fijo de movimientos que separan los grados de oportunidad individuales. La inicialización de la postura corporal y la longitud de la extremidad de las localizaciones articulares identificadas manualmente usando imágenes monoculares ha sido abordada en varios trabajos. Un método para inicializar automáticamente la estructura cinemática de la parte

superior del cuerpo ha sido investigado y utilizando la segmentación de movimiento de imágenes de video monoculares. P-resentaron un algoritmo de aprendizaje no supervisado que utiliza pistas características de puntos de secuencias de video monocular desordenadas para automatizar el proceso de desarrollo de modelos triangulares de cinemática de cuerpo entero.

La tabla 1 muestra una comparación entre los enfoques para el Modelo de Inicialización de acuerdo a varios autores.

1.1.3 Inteligencia artificial

La inteligencia artificial es parte de la teoría y el desarrollo de sistemas computacionales con la capacidad de poder desarrollar tareas que normalmente requieren la inteligencia humana, tales como la percepción visual, el reconocimiento de voz, la toma de decisiones (Sanlam, 2018).

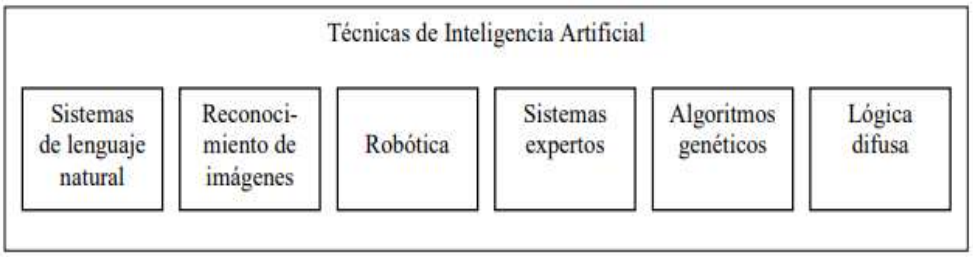


Figura 1. Subcampos de la Inteligencia Artificial

Fuente:(Sanlam, 2018)

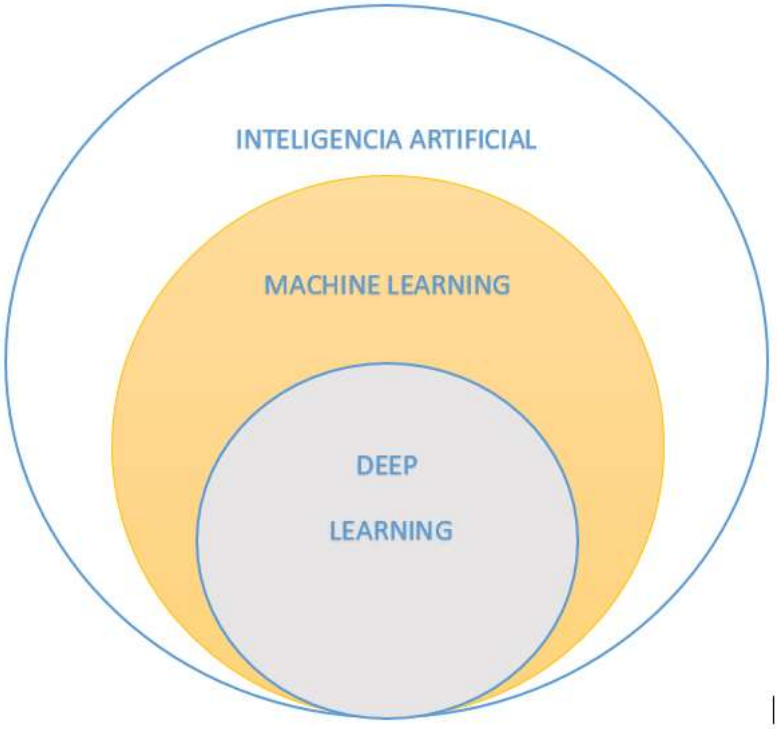


Figura 2 División de la inteligencia artificial

Fuente: ("Artificial Intelligence and Machine Learning for Dummies," 2019)

1.1.4 Redes neuronales

Los paradigmas de aprendizaje y procesamiento automático están inspirados en el funcionamiento del sistema nervioso humano. La red neuronal está compuesta por un conjunto de neuronas interconectadas entre sí mediante enlaces.

Cada neurona toma como una entrada las salidas las neuronas antecesoras, multiplica cada una de las entradas por un peso y mediante una función de activación calculan una salida esta salida es a su vez entrada de la neurona a la que precede. La unión de todas estas neuronas interconectadas es lo que compone una red neuronal artificial.

Estas redes no son más que redes interconectadas masivamente en paralelo de elementos simples (usualmente adaptativo) y con organización jerárquica, las cuales intentan interactuar con los objetos del mundo real del mismo modo que lo hace el sistema nervioso biológico (Calvo, 2017).

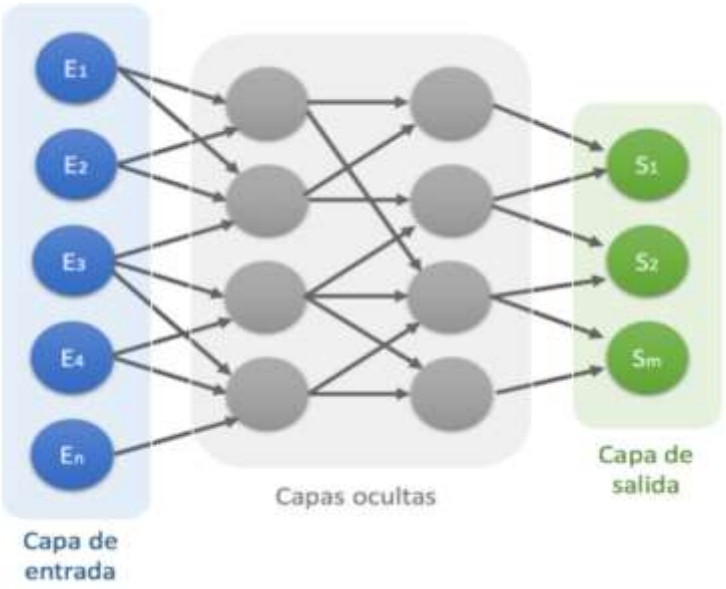


Figura 3. Red Neuronal Artificial

Fuente: (Calvo, 2017)

1.1.5 Redes neuronales artificiales

Las redes neuronales artificiales son métodos y de nuevos enfoques de inspiración biológica que permite que un algoritmo pueda aprender a partir de conjuntos de datos. Una red está compuesta por neuronas que procesan la suma de los datos de entrada y generan una salida entre 1 y 0.

El aprendizaje en la inteligencia artificial se produce al obtener una medida de la señal del error obtenido, comparando las predicciones de la red y los valores esperados, con la finalidad de ajustar los pesos Wi, para reducir gradualmente dicha medida de error (Martínez, 2018).

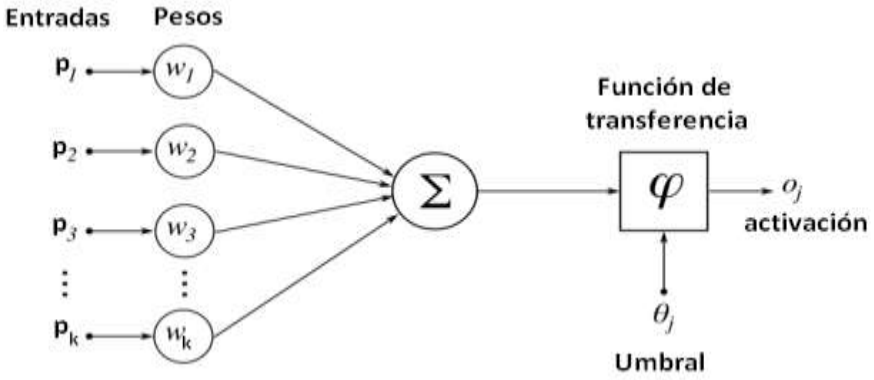


Figura 4. Diagrama de una neurona con la función de activación umbral

Fuente: (Kawaguchi, 2000)

1.1.6 Arquitectura de AlexNet

Esta arquitectura fue una de las primeras redes neuronales profundas en impulsar la precisión de la clasificación de ImageNet. Esta arquitectura está compuesto por 5 capas convolucionales seguidas por 3 capas completamente conectadas como se muestra en la figura (5) (Cobos & González, 2015)

Es la arquitectura menos profunda y así como la más antigua y fue la arquitectura ganadora de la competencia ILSVRC del 2012 marcando un hito al superar el estado del arte establecido hasta el momento en la base de datos ImageNet de forma drástica. A pesar de ser la arquitectura más antigua de las estudiadas sigue siendo empleada por su gran funcionamiento (Amador & Baumela, 2017).

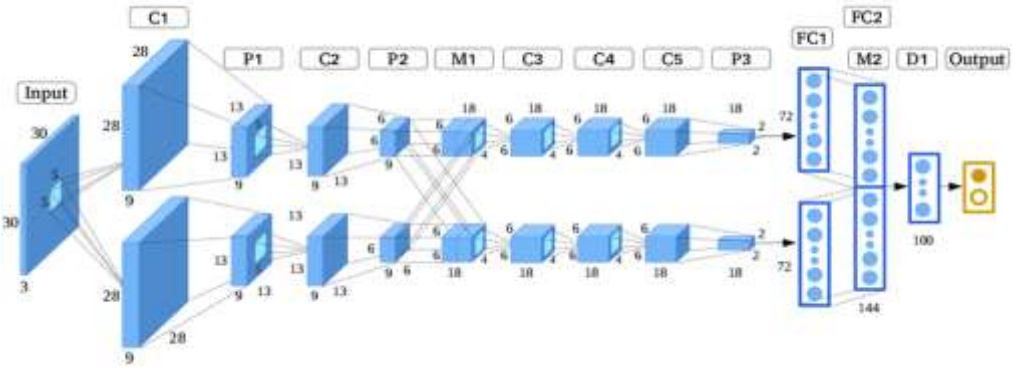


Figura 5. Ilustración de la Arquitectura AlexNet

Fuente: (Krizhevsky, Sutskever, & Hinton, 2012)

1.1.7 VGG Face

Esta red es una adaptación genérica de 16 capas VGG lo particular de esta red y lo que hace interesante para nuestro estudio es que esta entrenada específicamente para caras. El propósito es etiquetar la identidad de una persona entre un total de 2622 identidades. (Domínguez, & Baumela, s.f).

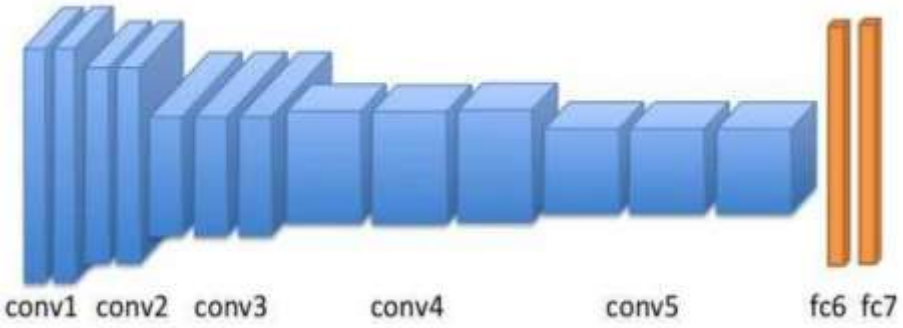


Figura 6. Arquitectura de una Red VGG-Face

Fuente: (Nakada, Wang, & Terzopoulos, 2017)

1.1.8 VGG – 16

La arquitectura VGG-16 ha sido ampliamente utilizada en visión por computador en los últimos años donde se compone una apilada capa convolucionales y de agrupación máxima es utilizada por ser una arquitectura de 16 capas más pequeña y por lo tanto más rápida y conocida como VGG-16.(Ferguson, 2017).

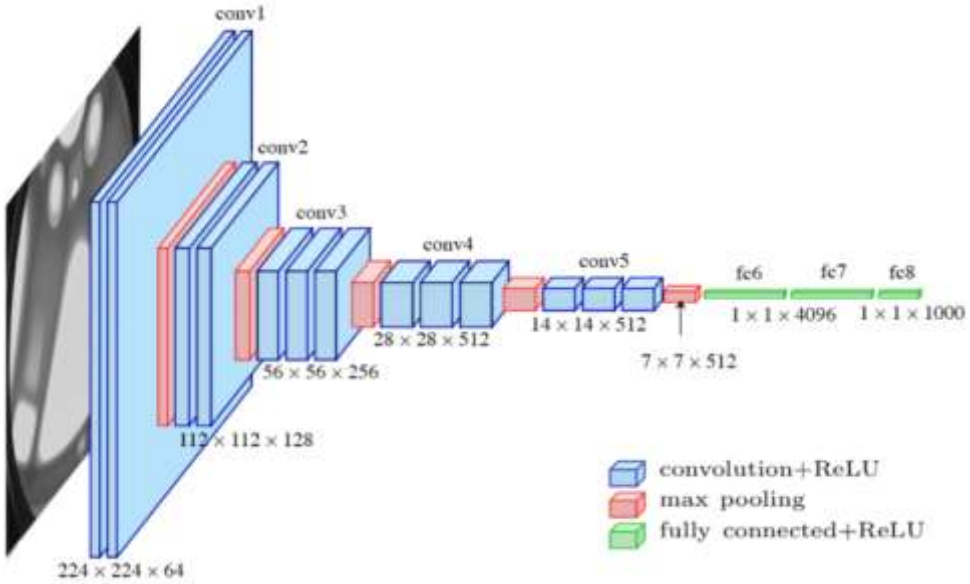


Figura 7. Arquitectura VGG-16

Fuente:(Ferguson, 2017)

1.1.9 Redes neuronales recurrentes

Una red neuronal recurrente no tiene una estructura de capas definida, sino que permiten conexiones arbitrarias entre las neuronas, incluso pudiendo crear ciclos, con esto se consigue crear la temporalidad, permitiendo que la red tenga memoria. Las redes neuronales recurrentes son muy potentes para todo lo que tiene que ver con el análisis secuencias, como puede ser el análisis de textos, sonido o video (Calvo, 2017).

"Existe multitud de tipos de redes neuronales dependiendo del número de capas ocultas y la forma de realizar la retro propagación." (Calvo, 2017).

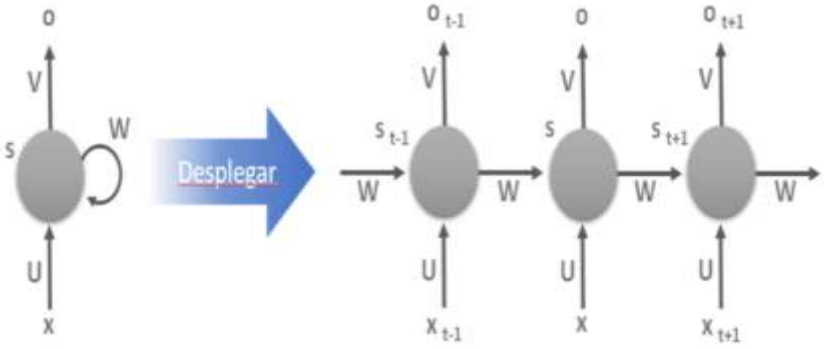


Figura 8. Red neuronal recurrente

Fuente:(Calvo, 2017)

1.1.10 Máquinas de soporte vectorial

Las máquinas de soporte vectorial (SVM) es una técnica de clasificación donde fueron introducidos por vladimir Vapnik y su grupo de trabajo. La primera mención fue en 1979 donde el principal documento fue publicado en 1995. (Ledezma Willmar, 2012).

Las máquinas de soporte vectorial (Support Vector Machine) son un nuevo sistema de aprendizaje el cual ha tenido un desarrollo muy significativo en los últimos años en la generación de nuevos algoritmos como en las estrategias para su implementación. SVM es un sistema de aprendizaje basado en el uso de un espacio de hipótesis de funciones lineales en un espacio de mayor dimensión inducido por un kernel, en el cual las hipótesis son entrenadas por un algoritmo. (Resendiz, 2006)

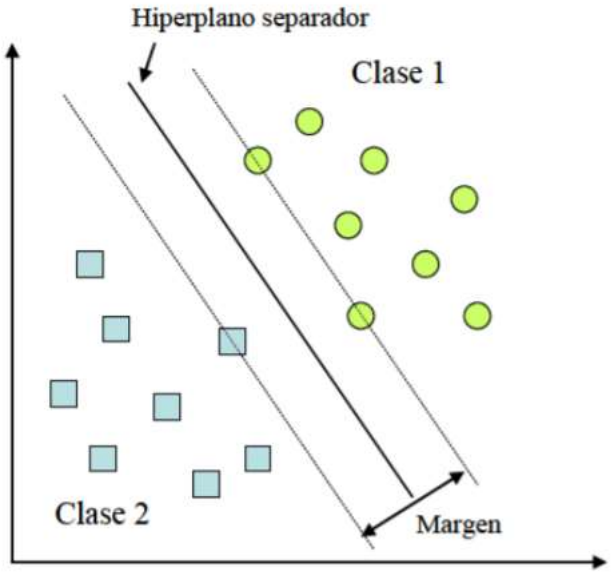


Figura 9. Separación de datos mediante SVM

Fuente:(Otero, 2012)

1.1.10.1 Caso linealmente separable

Para este caso linealmente separable, las MVS conforman hiperplanos que separan los datos de entrada en dos subgrupos que poseen una etiqueta propia. En medio de todos los posibles planos de separación de las dos clases etiquetadas como {-1, +1}, existe sólo un hiperplano de separación óptimo, de forma que la distancia entre el hiperplano óptimo y el valor de entrada más cercano sea máxima (maximización del margen) con la intención de

forzar la generalización de la máquina que se esté construyendo como se muestra en la siguiente función. (Ledezma Willmar, 2012).

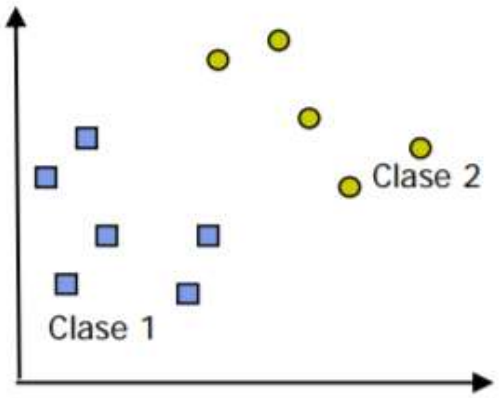


Figura 10. Caso linealmente separable.

Fuente: (Ech everry & Urueña, 2008)

1.1.10.2 Caso no linealmente separable

Para tratar con datos que no son linealmente separables, el análisis previo puede ser generalizado introduciendo algunas variables no-negativas $\xi \geq 0$ de tal modo que es modificado.(Universidad Tecnológica de Pereira., 2008).

$$y_i(w \cdot z_i + b) \geq 1 - \xi_i, \qquad i = 1, \ldots, l.$$

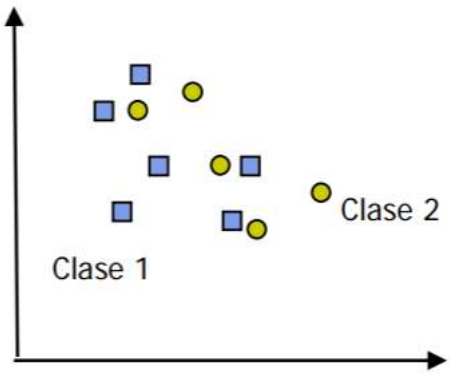


Figura 11. Caso no linealmente separable

Fuente: (Echeverry & Urueña, 2008)

1.1.11 Redes neuronales convolucionales

Las redes neuronales convolicionales son una extencion del perceptron multicapa, con la diferencia de que las CNNs realizan convoluciones entre los parámetros y los datos de la red, donde son apropiadas para aplicaciones en las que los datos se encuentran en forma de una rejilla, como matrices, a diferencia de una Multi Layer Perceptron (MLP), las CNN procesan imagens por secciones y han tenido un notable éxito (Vizcaya, Albino, & Lazcano., 2017)

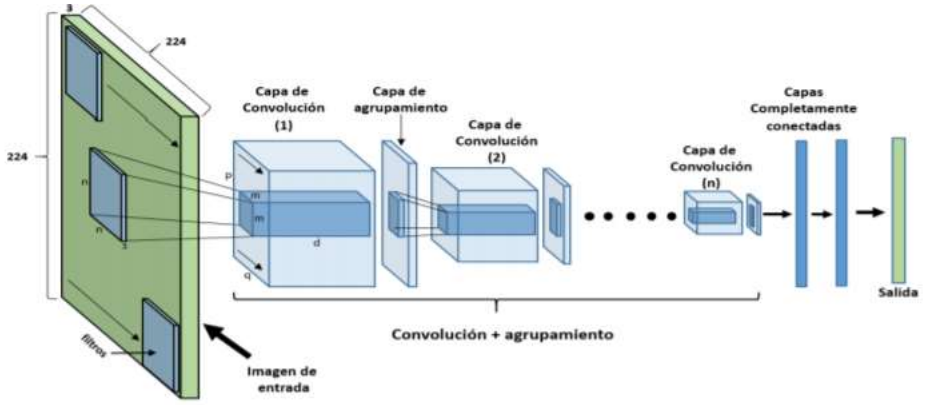


Figura 12. Arquitectura básica de una red neuronal convolucional

Fuente: (Vizcaya, Albino, & Lazcano., 2017)

Tabla 1

Comparación entre diferentes redes neuronales convolucionales

	# capas de convolucion	MAACs ($X10^6$)	Parámetros ($X10^6$)	Activación	ImageNet Top-5 error %
Alexnet	5	1140	62.4	2.4	19.7
Network in Network(2013)	12	1100	7.6	4.0	19.0
VGG-16	16	15470	138.3	29.0	8.1
GoogLenet(2015)	22	1600	7.0	10.4	9.2
ResNet(2015)	50	3870	25.6	46.9	7.0
Inception v3(2016)	48	5710	23.8	32.6	5.6

Fuente: (Letelier, 2006)

1.1.12 Estructura de una red neuronal convolucional

1.1.12.1 Capas de convolucion

En este concepto se basa en el fundamento de las CNN. La operación de la convolucion .

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n)$$

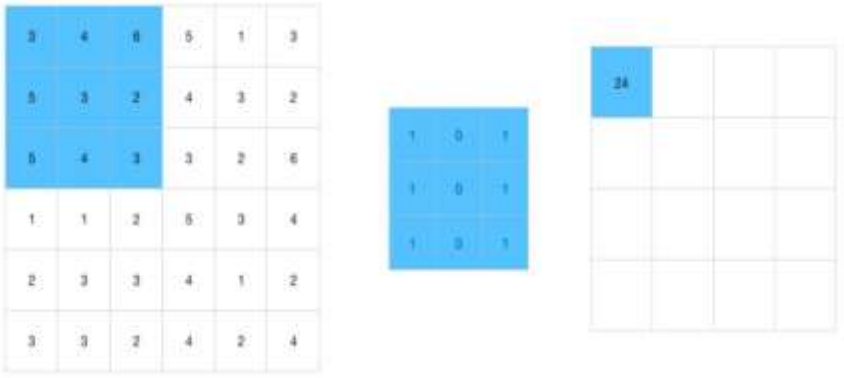


Figura 13. Capa de la convolucion

Fuente: (Rodríguez, 2018)

1.1.12.2 Capas de agrupamiento o pooling

El propósito de esta capa es reducir el tamaño de los datos progresivamente, con el fin de reducir los parámetros a tratar y consecuentemente la cantidad de cálculos, estas capas solo reducen el tamaño espacial de los datos utilizando la operación MAX que consiste en dividir los datos en secciones, para extraer los valores máximos de cada sección, discriminando los demás.(Vizcaya, 2017)

$$W_2 = \frac{(W_1 - F + 2P)}{S} + 1$$
$$H_2 = \frac{(H_1 - F + 2P)}{S} + 1$$
$$D_2 = K$$

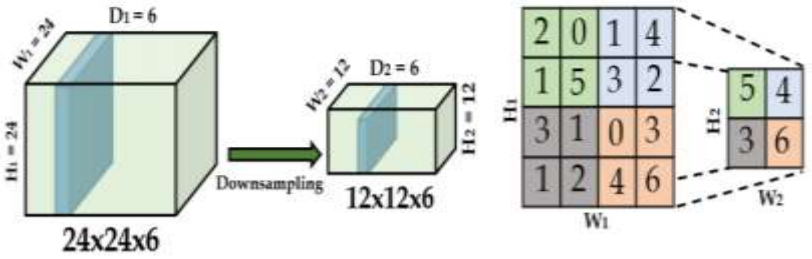


Figura 14. Capa de pooling o agrupamiento

Fuente: (Vizcaya, Albino, & Lazcano., 2017)

1.1.12.3 Capa completamente conectada

En esta capa, cada nodo se encuentra completamente conectado a las salidas de la capa anterior. Cada nodo hace una suma ponderada de sus parámetros por el valor de entrada, además la entrada independiente. Con ello el que tenga mayor puntuación, será el que tenga mayor probabilidad. En si hacen una clasificación indicando la probabilidad de cada clase. (Vizcaya., 2017)

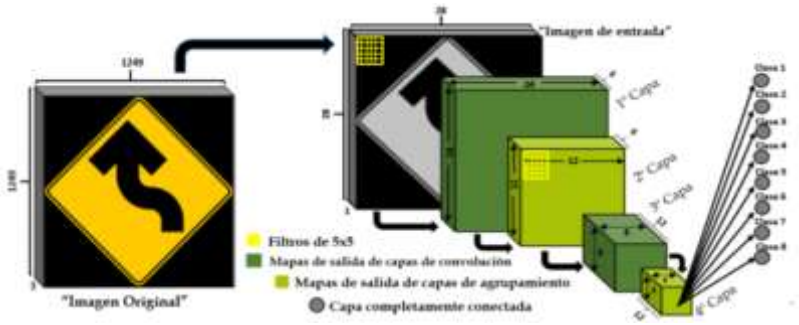


Figura 15. Arquitectura implementada completamente conectada

Fuente: (Vizcaya, Albino, & Lazcano., 2017)

1.2 Antecedentes

En este capítulo se revisan los trabajos de investigación recientes enfocados en detección de eventos en videos. Se describe los trabajos desarrollados y algoritmos usados y los resultados obtenidos.

(Bulat & Tzimiropoulos, 2017) Proponen una técnica de estimación de la pose humana haciendo uso de las redes neuronales convolucionales. Y la principal contribución es una arquitectura CNN en cascada diseñada específicamente para el aprendizaje de las

relaciones entre las partes del cuerpo. Para tal se propone la detección de cascada CNN y seguida por la regresión. La primera parte muestra salidas en cascada, mapas de calos de detección de partes y la segunda parte realiza la regresión en estos mapas de calor.

(Cao, Simon, & Shih., 2018) Propone un nuevo enfoque para detectar posturas de varias personas en una secuencia de videos/imágenes haciendo uso de los puntos de afinidad (PAF), para aprender a asociar partes del cuerpo de las personas. La arquitectura está diseñada para aprender conjuntamente la ubicación de las partes y su asociación a través de dos ramas des mismo proceso de predicción secuencial. Donde el método se ubicó en el primer lugar en el desafío inaugural de puntos clave de COCO 2016 y supera significativamente el resultado del estado de la técnica anterior en el punto de referencia MPII Multi personas tanto en rendimiento como en eficiencia.

En España (Pantrigo, 2018) proponen como objetivo de establecer su rendimiento bajo diferentes condiciones experimentales se plantea la opción de encontrar soluciones de alta calidad en tiempos menores, donde se propone la hibridación entre algoritmos de estimación secuencial y optimización combinatoria. En concreto, se han desarrollado los algoritmos denominados filtros de partículas con re encadenamiento de trayectorias y el filtro de partículas con búsqueda dispersa ambos algoritmos se han aplicado a la determinación de postura del cuerpo humano en diferentes situaciones, como carrera, saltos, movimientos, etcétera.

(Realpe & Vintimilla, 2009) El artículo de investigación presentan una técnica para la detección y seguimiento de personas en movimiento a partir de secuencias de videos. Haciendo uso de algoritmos de diferenciación temporal y sustracción de fondos y un proceso de filtrado a través de operadores morfológicos aplicado a objetos detectados a modo de eliminar ruidos y generar únicamente siluetas en movimiento y representando a personas. Al final d todo el procedimiento se obtiene un listado que representa el desplazamiento de ambos grupos de puntos a través del tiempo y poder realizar la interpretación de acciones o de actividades humanas en un futuro trabajo.

(Manejes, 2017) el trabajo de investigación propone un nuevo modelo de detección y localización de eventos anómalos en áreas peatonales, con el objetivo de diseñar un

algoritmo que permita detectar eventos anómalos mediante el uso de la información del movimiento y la apariencia, a diferencia de los métodos de la literatura, el modelo propuesto proporciona una solución general para detectar eventos anómalos tanto globales como locales, además en la etapa de la detección se presenta problemas de perspectiva debido a que los objetos cercanos a la cámara parecen ser grandes, mientras que los objetos alejados a la cámara parecen ser pequeños para abordar estos problemas también se propone la clasificación por región.

(Guler, Neverova, & Kokkinos, 2018) El presente trabajo de investigación establece correspondencias densas entre una imagen RGB y una representación basada en superficie de cuerpo humano, tarea a la que es referida como pose humana donde se hace una estimación y recopilación de correspondencias densas de 50,000 personas que aparecen en los DATASETS COCO. Haciendo uso del conjunto de datos para entrenar. Donde se tiene que mejorar la efectividad del conjunto de entrenamientos, entrenando una red de pintura que puede llenar la falta de valores de verdad en el campo de informes de mejoras con respecto a los resultados que se puedan lograr. Donde también se experimenta con redes completamente convulsiónales y modelos mejorando aún más con precisión a través de la cascada donde obtienen resultados altamente precisos en múltiples cuadros por segundo en un solo GPU.

(Osorio & Holguín, 2015) relata la investigación del desarrollo de una herramienta para la detección y clasificación de objetos cuya morfología es el factor discriminante más importante. Donde presenta una metodología basada en el descriptor más conocido como histogramas de gradientes orientados, o HOG *Histogram of Oriented Gradients*, este descriptor alimenta una máquina de vectores de soporte que permite realizar la clasificación de objetos deseado. Donde se propone una reducción sistemática de la dimensionalidad de HOG mediante la identificación de bloques de descriptor de discriminante para mejorar la eficiencia general del sistema.

(Ramírez et al., 2013) indican en la investigación y presentan los resultados de investigación obtenidos en la etapa del diseño de un algoritmo que permite la detección y seguimiento de un objeto mediante grabaciones de videos. El algoritmo de diseño e implemento en MatLab los videos fueron facilitados por el centro de investigación

Apícola Tropical. El principal resultado que se tiene es la implementación de un programa capaz de detectar y registrar el movimiento de objetos. Lo cual es algo innovador y útil para estudiar el comportamiento de muchos objetos.

(Ferrari, Jimenez, & Zisserman, 2008) propone la técnica de como estimar la pose humana 2d como una configuración espacial de las partes del cuerpo en la tomas de video. Por lo que se propone es un enfoque que reduce progresivamente el espacio de búsqueda de las partes del cuerpo, para mejorar en gran medida las posibilidades de que la estimación de postura tenga éxito. Por otro lado, también se propone un modelo espacio temporal que abarca múltiples marcos para refinar las estimaciones de posturas de marcos individuales.se demuestra la estimación de pose en la parte superior del cuerpo mediante una extensa evaluación de más de 70000 fotogramas de cuatro episodios de series de video.

(Eichner & Ferrari, 2012) realiza la demostración de su técnica de Human Position Co estimación PCE en dos tipos de aplicaciones. El primero estimar la pose de la persona que realiza la misma actividad sincrónica, como ejercicios aeróbicos, bailes en grupo. La segunda aplicación es el aprendizaje de las poses de prototipos que caracterizan una clase de pose directamente desde un motor de búsqueda de imágenes como la postura de “loto” y mostramos que PCE conduce una mejor estimación de postura en imágenes.

(Fuentes &Velastin, 2004) Proponen una aplicación en el campo de la vigilancia automática basado en el uso de un algoritmo que permite la localización y el seguimiento de múltiples objetos con características de los objetos detectados, sus posiciones y trayectorias y es posible obtener información como para determinadas acciones. Y en lo general poder ayudar a los operadores a optimizar el uso de los recursos.

(Barbuzza et al., & 2017) El trabajo presenta resultados preliminares de un algoritmo de detección y eliminación de sombras, en secuencias de video. Y se propone a partir de la base sustracción de fondo *Visual Background Extraction (ViBE)* que identifica zonas de movimiento aplicar un post-procesamiento para separar los pixeles del objeto real, además de realizar análisis de similitudes entre el cuadro actual y el modelo de fondo. El

algoritmo se puede aplicar para poder detectar personas en aplicaciones en seguridad ciudadana, análisis deportivo entre otros.
Los resultados obtenidos en la detección de objetos se muestran que es factible recortar la sombra con alta tasa de acierto.

(Pantrigo, s.f.) Empleando el filtro de Kalman que es una técnica recursiva para determinar los parámetros correctos de un sistema, el filtro de kalman se utiliza en situaciones donde un proceso continuo es muestreado en intervalos de tiempo y tiene de especial interés en el seguimiento de objetos en secuencia de imágenes. Cuyo objetivo principal de este proyecto recae en la aplicación del filtro de kalman al seguimiento de objetos de imágenes. Como resultados se consiguió implementar una aplicación de filtro de Kalman mediante las técnicas de visión artificial.

(Gervasoni, Damato & Barbuzza, 2016) el artículo de investigación propone una variante de algoritmo para el estudio de objetos en situaciones particulares a partir de video vigilancia, donde se presenta una implementación GPU utilizando OpenCL para lograr un análisis de sustracción de fondos en tiempo real.

(Hernández, Reyes & Ponce., 2012) Proponen una metodología de segmentación humana Spatio-Temporal GrabCut totalmente automática que combina el seguimiento y la segmentación. La inicialización de GrabCup se realiza mediante una detección de sujeto basada en HOG, detección de rostros y modelos de color de piel. La información incluye el agrupamiento de Mean Shift y el histórico Gaussian Mixture Models. Y la pose se obtiene al combinar la segmentación humana y modelos de aspecto activo. Los resultados de un conjunto de datos públicos y en un nuevo conjunto de datos humanos muestran una sólida segmentación y recuperación tanto de la cara como la postura.

(Raúl & Molina, 2010) España, presentan una técnica sobre esqueletización que permite obtener, a partir de un objeto digital dado, otro que contiene la misma información topológica, nos vamos a centrar en las técnicas desarrolladas por G. Bertrand por su similitud de técnicas homológicas de procesamiento de imágenes, desarrollando un algoritmo de esqueletizacion a nivel de complejos celulares. Es así que presentan un algoritmo de esqueletizacion utilizando el marco de computación natural denominado computación con membranas.

(Carnegie, 2016) Presenta un enfoque para detectar eficientemente la pose 2D de varias personas en una sola imagen donde el enfoque utiliza una representación paramétrica la que definimos como afinidad para aprender asociar partes del cuerpo humano. Independientemente el número de personas en la imagen la arquitectura está diseñada para aprender conjuntamente las ubicaciones y partes del cuerpo. El método propuesto se colocó primero en los desafíos de puntos clave de COCO 2016 donde se obtiene resultados de referencia de la persona tanto en rendimiento como eficiencia.

(Realpe, Vintimilla, Romero, & Remagnino, 2018) El artículo de investigación propone una técnica para la detección y seguimiento de personas en movimiento a partir de secuencia de video. Haciendo uso de algoritmos de diferenciación temporal y sustracción de fondo de los objetos detectados y a modo de eliminar el ruido y generar únicamente siluetas de movimiento representando a personas. Y al final del procedimiento como resultados se obtiene un listado que representa el desplazamiento de ambos grupos de puntos a través del tiempo para la interpretación de actividades y acciones humanas.

(Cobos & González, 2018) Leganés-España, con este trabajo se pretende conseguir y calcular las trayectorias seguidas por personas dentro de un campo de visión de cámara. Puesto que hace uso de técnicas como la detección de movimiento que son usadas por múltiples aplicaciones como en casos de seguridad, entrenamiento (Kinect), etcétera. El objetivo es que se realizara un cálculo de la trayectoria seguida por personas gracias a una cámara, un sensor de distancia incluido a personas y un ordenador para realizar el procesado.

(Sidenbladh, 2004) En esta investigación presentan un método para detección de humanos, La idea es detectar patrones de movimiento humano, una máquina de vectores de soporte esta entrenada por patrones de flujo óptico denso que se originan en humanos. Este SVM entrenado es el núcleo de un algoritmo de detección humana que busca imágenes de flujo óptico para patrones de movimiento similares a los humanos.

(Simon, Joo & Matthews, 2017) Presentan un nuevo enfoque que hace uso de las multi cámaras para entrenar detectores de grano fino para los puntos clave que son propensos a

oclusión, como las articulaciones de la mano, este procedimiento es denominado bootstrapping multivision, se usa para producir producir etiquetas ruidosas en múltiples vistas de la mano, las detecciones ruidosas se triangulan en 3D usando la geometría de vistas múltiples. El método se utiliza para entrenar un detector de punto clave donde el detector de punto clave resultante se ejecuta en tiempo real en imágenes RGB.

(Wei & Ramakrishna, 2016) La investigación de máquinas de postura proporciona un marco de predicción secuencial del aprendizaje de modelos espaciales, donde se demuestra el diseño sistémico de como las redes convulsiónales se pueden incorporar en el marco de la máquina de pose para aprender las características de la imagen y los modelos espaciales dependientes de la imagen para la tarea de estimación de postura. La contribución de la investigación es modelas implícitamente las dependencias de largo alcance entre varias tareas de predicción estructuradas, como la estimación de postura articulada logrando esto mediante una arquitectura secuencial compuesta por redes convolucionales.

(Dalal, 2005) Presenta un estudio basado en características para objetos visuales en reconocimiento, adoptando detección humana lineal basada en SVM como un caso de prueba donde hacen muestras experimentales de descriptores de histogramas de gradiente orientado (HOG) donde significa que puede superar ampliamente los conjuntos de características existentes para la detección humana, el enfoque proporciona una separación casi perfecta en MIT original de base de datos peatonal por lo cual presentan un conjunto de datos más desafiantes que contiene 1800 imágenes humanas anotadas con una amplia gama de variaciones de pose

CAPÍTULO II

PLANTEAMIENTO DEL PROBLEMA

2.1 Planteamiento del problema

La detección del comportamiento humano ha sido estudiada bajo distintas perspectivas en una amplia variedad de disciplinas, como la psicología, la biomecánica, la computación gráfica y visión computacional. En este contexto, se ha definido que una primitiva de acción es un movimiento anatómico que puede describirse movimientos en el nivel de las extremidades. Una acción consiste en primitivas de acción y describe un movimiento, posiblemente cíclico, de todo el cuerpo.

Detectar el comportamiento humano utilizando inteligencia artificial, visión computacional y otras áreas de investigación es un campo relativamente reciente, y actualmente en constante investigación.

La principal pregunta de investigación bajo este contexto se puede caracterizar de la siguiente manera: dada una sucesión de imágenes, donde aparecen personas, se puede detectar que estas están realizando alguna acción, adicionalmente se puede resaltar con un marco para distinguir: quién o qué objeto está realizando la acción y qué acción se realizó. Gracias a investigaciones sobre esta área, será posible aplicar dichos conocimientos en una amplia variedad de actividades humanas, como los deportes, la vigilancia visual, la detección de incidentes y accidentes.

La taxonomía se divide en tres temas principales: técnicas de detección, conjuntos de datos y aplicaciones. Las técnicas de detección fueron divididas en cuatro categorías (inicialización, seguimiento-tracking, estimación y reconocimiento). La lista de conjunto de datos, corresponde a la secuencia de imágenes y videos utilizados para aplicarlos en las pruebas en los métodos utilizados. Finalmente se identificaron varias áreas de

aplicación, incluyendo detección humana, detección de actividad anormal, reconocimiento de acciones, modelado de jugadores y detección de peatones.

2.2 Problema de investigación

Dada la inseguridad que actualmente vivimos, y visto que existe la disponibilidad de cámaras de video para video vigilancia, es posible utilizar estos recursos para poder analizar estas imágenes (video) y poder interpretar posibles movimientos y acciones y prevenir acontecimientos indeseados.

2.3 Resultados esperados de la investigación

RE1. Una aplicación (software) para la detección automática de eventos inusuales captados mediante cámaras de video vigilancia.
RE2. Con la finalidad de validar los resultados de la aplicación desarrollada se emplearán imágenes/video captados en el centro comercial Nro. 2 de la ciudad de Juliaca.

2.4 Intención de la investigación

Desarrollar un sistema para detección de eventos inusuales en imágenes y videos captados por cámaras de video vigilancia.

2.5 Alcance

El presente proyecto de investigación abarca conceptos y búsqueda de investigaciones desarrolladas y centradas en la identificación y detección de comportamientos humanos en video. Se harán uso de las técnicas y métodos que utilizan las redes neuronales para así poder extraer las características y su respectiva clasificación.
Para la identificación de comportamiento y acciones humanas se empleará la arquitectura de una red convolucional que consta de dos etapas la extracción de las partes y los puntos de afinidad.

2.6 Justificación

Actualmente vivimos en una sociedad con violencia, usualmente son accidentes, aunque también es común ver eventos y acontecimientos de delincuencia. Ambos pueden ocasionar pérdidas irreparables como pérdidas humanas. Estas pérdidas pueden ser inmediatas o a consecuencia, aunque en una gran mayoría de casos las perdidas vienen seguidas del acto de violencia, siendo que estas pueden ser evitadas si es que se atiende de manera oportuna. Para este propósito la visión computacional ha venido enfocándose

en plantear diversas alternativas en base al análisis de imágenes y video (secuencia de imágenes), considerando que las ciudades modernas actualmente vienen instalando cámaras de vigilancia distribuidas en diversos lugares, así como propietarios de inmuebles lo hacen también para poder mejorar su seguridad. Es así, que es posible aprovechar estos recursos y plantear una alternativa de aplicación que permita apoyar a la seguridad por medio del análisis automático de imágenes y video captados por cámaras de vigilancia.

2.7 Objetivos

2.7.1 Objetivo general

Implementar una aplicación computacional para "detección de eventos inusuales en imágenes y video de cámaras de vigilancia".

2.7.2 Objetivos específicos

- Analizar y procesar imágenes y video
- Detectar patrones en imágenes y video
- Automatizar y alertar automáticamente

CAPÍTULO III

MATERIALES Y MÉTODOS

3.1 Metodología

3.1.1 Diseño de la investigación

Esta investigación según su finalidad es "Investigación aplicada", según los medios utilizados es "experimental" y según el nivel de conocimientos es "descriptivo". Por ser una investigación en el campo de las ciencias de la computación no se adapta al uso de población y muestra.

3.1.2 Metodología de desarrollo

Para lograr llegar al objetivo planteado se debe de identificar un modelo de red neuronal convolucional para su clasificación de características y partes del cuerpo humano e identificar la base de datos de entrenamiento y evaluar dicha arquitectura de estimación de pose en video.

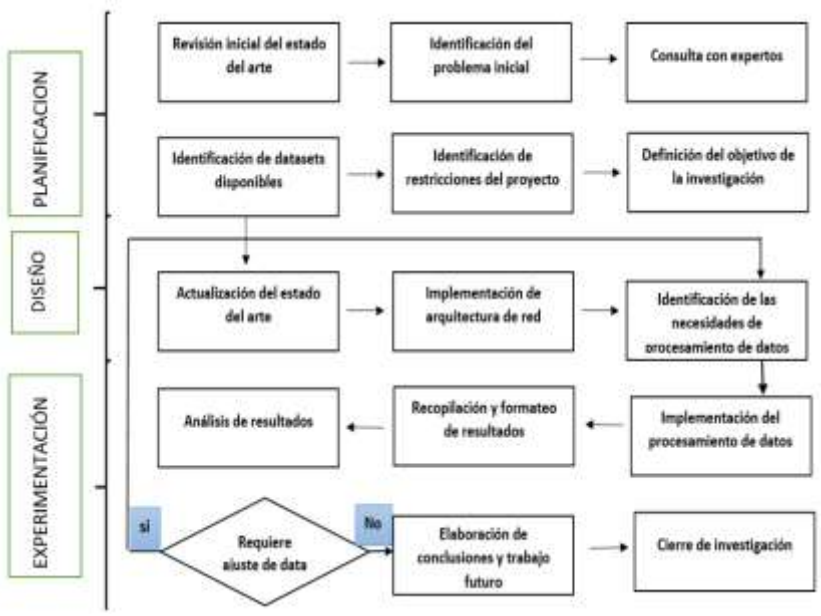


Figura 16. Metodología utilizada

3.2 Programación extrema (Xp)

Es una metodología ágil centrada en potenciar las relaciones interpersonales como una clave del éxito en desarrollo de software. XP se basa en la realimentación continua entre cliente y equipo de desarrollo, comunicación entre todos los participantes simplicidad en las soluciones implementadas. XP se define como especialmente para proyectos con requisitos muy cambiantes, donde existe un alto riesgo técnico.(Letelier, 2006)

3.2.1 Procesos de programación extrema

Un proyecto de programación extrema tiene éxito cuando el cliente selecciona el valor del negocio a implementar basado en la habilidad en equipo para medir funcionalidades que puedan entregar a través del tiempo. El ciclo del desarrollo consiste en las siguientes fases. (Letelier, 2006)

- Exploración
- Planificación de entrega
- Iteración
- Producción-
- Mantenimiento
- Muerte del proyecto

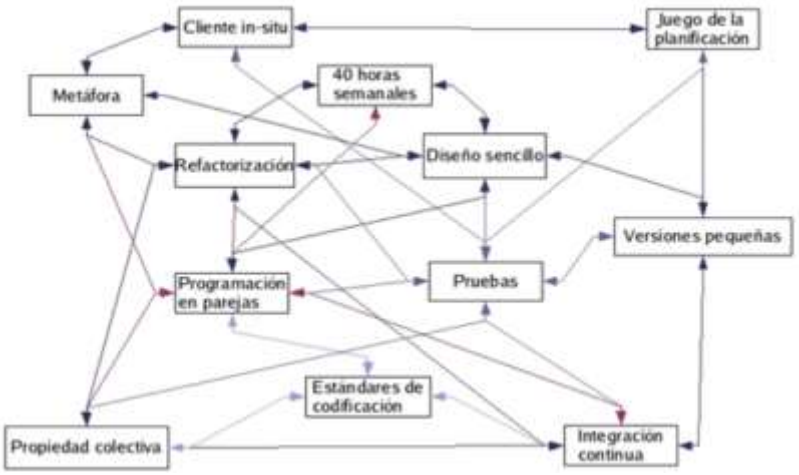


Figura 17. Estructura de una metodología XP

Fuente: (Letelier, 2006)

Tabla 2

Comparación de metodologías agiles

	CMM	ASD	CRYSTAL	FDD	LD	SCRUM	XP
Sistema algo cambiante	1	5	4	3	4	5	5
Colaboración	2	5	5	4	4	5	5
Resultados	2	5	5	4	4	5	5
Simplicidad	1	4	4	5	3	5	5
Adaptabilidad	2	5	5	3	4	4	3
Excelencia técnica	4	3	3	4	4	3	4
Practica de colaboración	2	5	5	3	3	4	5
Media CM	2.2	4.4	4.4	3.8	3.6	4.2	4.4
Media Total	1.7	4.8	4.5	3.6	3.9	4.7	4.8

Fuente: (Letelier, 2006)

3.3 Máquinas de postura convolucionales

El método empleado para la detección y estimación de postura para eventos es la "Estimación de Múltiples Personas Usando Campos de Afinidad de Partes (PAF), una técnica de máquinas de postura que permite detectar los puntos de interés (articulaciones) para luego emplear la técnica de "campos de afinidad de partes" para tal propósito se aplica una red neuronal convolucional (CNN) (Wei, Ramakrishna, & , 2016)

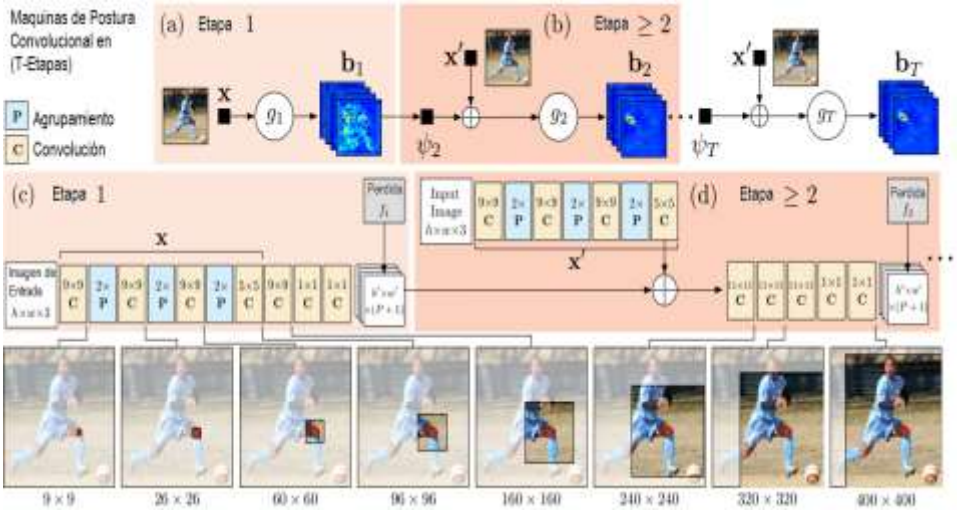


Figura 18. Técnica de máquinas de posturas convolucional

Fuente: (Wei & Ramakrishna, 2016)

Donde $\mathbf{G}_i$ es un clasificador, $\mathbf{G}_{i\ nos}$ permitirá encontrar los puntos de interés, esencialmente de las articulaciones grandes del esqueleto de una persona. Rodilla, cintura, codo, hombro, cuello, y la cabeza la misma técnica es empleada para la detección de los dedos de las manos y partes del rostro.

Aplicando la técnica de MPC (Máquinas de Postura Convolucional), en la primera etapa se obtienen los puntos de interés deseados como señales fuertes, aunque es posible que no sea posible resaltar algunos de difícil detección (codo derecho), pero empleando una técnica de correspondencia e interpolación en relación al cuello y cabeza puede ser estimado en las siguientes etapas.

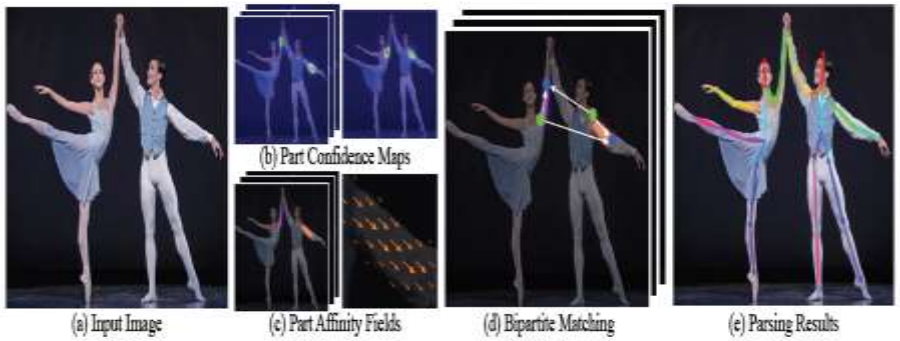


Figura 19. Toma de imagen completa en 2 ramas para predecir los mapas de confianza

Fuente: (Cao, Simon & Wei, 2018)

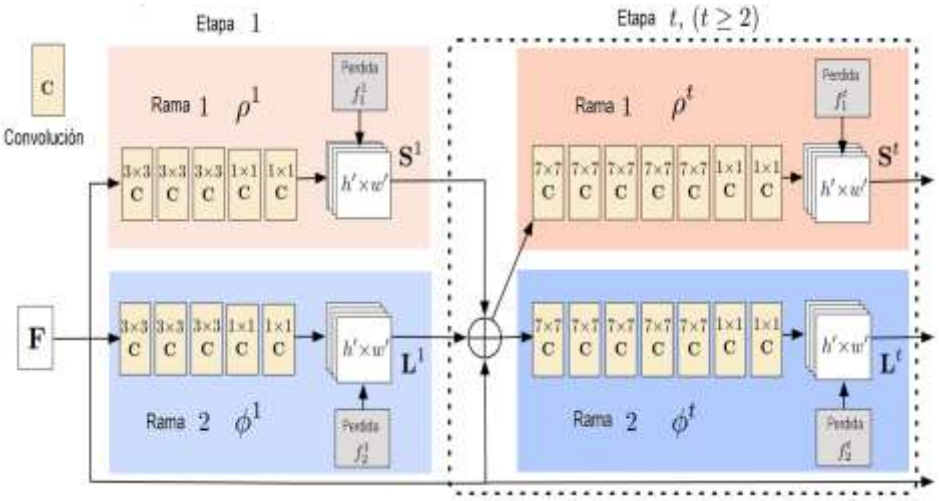


Figura 20. Arquitectura modificada en 2 ramas " convolucional multi-etapa "

Fuente:(Wei, Ramakrishna & Kanade, 2016)

Adicionalmente el método se apoya en el uso de un grafo de correspondencia para poder articular el cuerpo, empleando los puntos y partes de interés previamente obtenidos por la máquina de postura convolucional.

3.3.1 Detección y asociación simultanea

La arquitectura mostrada en la figura (21) predice simultáneamente mapas de confianza de detección de campos de afinidad que codifican la asociación de parte a parte. La red se divide en dos ramas la rama superior que se muestra de color beige, predice los mapas de confianza y la rama inferior que se muestran en azul predice los campos de afinidad cada rama es una arquitectura de predicción iterativa. La imagen se analiza primero mediante la red convolucional inicializado por las primeras 10 capas de VGG-19 que genera un conjunto de mapas de confianza de detección y un conjunto de campos de afinidad.

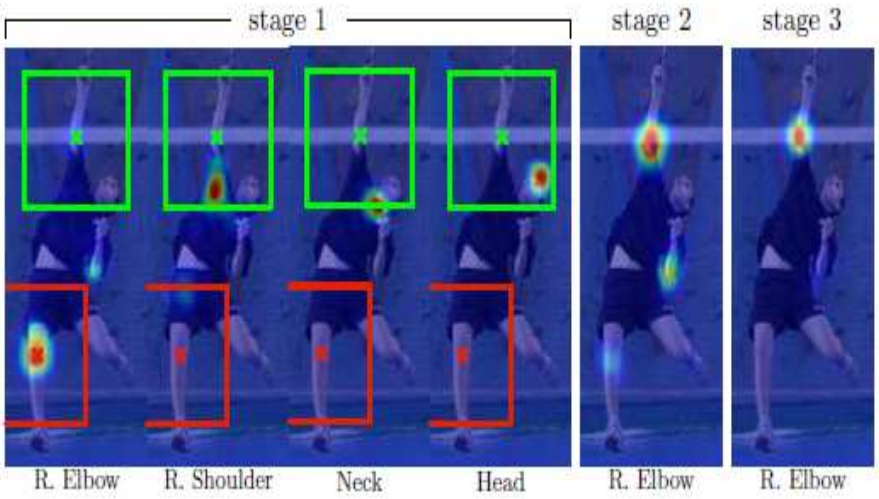


Figura 21. Contexto espacial de mapas de puntos principales

Fuente: (Wei & Ramakrishna, 2016)

3.3.2 Estimación de postura

La estimación de la postura alude a la metodología de evaluación de la disposición de la estructura de articulación cinemática o esquelética subyacente de un individuo. Los algoritmos de estimación de la posición se pueden dividir en tres categorías basadas en el uso del modelo humano. (Pantrigo Fernández, n.d.) Modelo libre-cuando no se utiliza ningún modelo anterior y la mayoría de las técnicas rastrea las partes del cuerpo en 2D o mapa de las secuencias 2D de

las percepciones de los cuadros en una pose en 3D. Uso indirecto del modelo-cuando se utiliza un modelo anterior dentro de la estimación de la postura (por ejemplo, etiquetado de la parte del cuerpo humano utilizando relaciones de aspecto entre miembros o distinción de postura).

Uso directo del modelo-cuando se utiliza una representación geométrica 3D explícita de la forma humana y de la estructura cinemática para reproducir la postura; La mayoría de los modelos directos utilizan estrategia de análisis por síntesis para mejorar la cercanía entre la proyección del modelo y las imágenes observadas. (Wei & Ramakrishna, 2016)

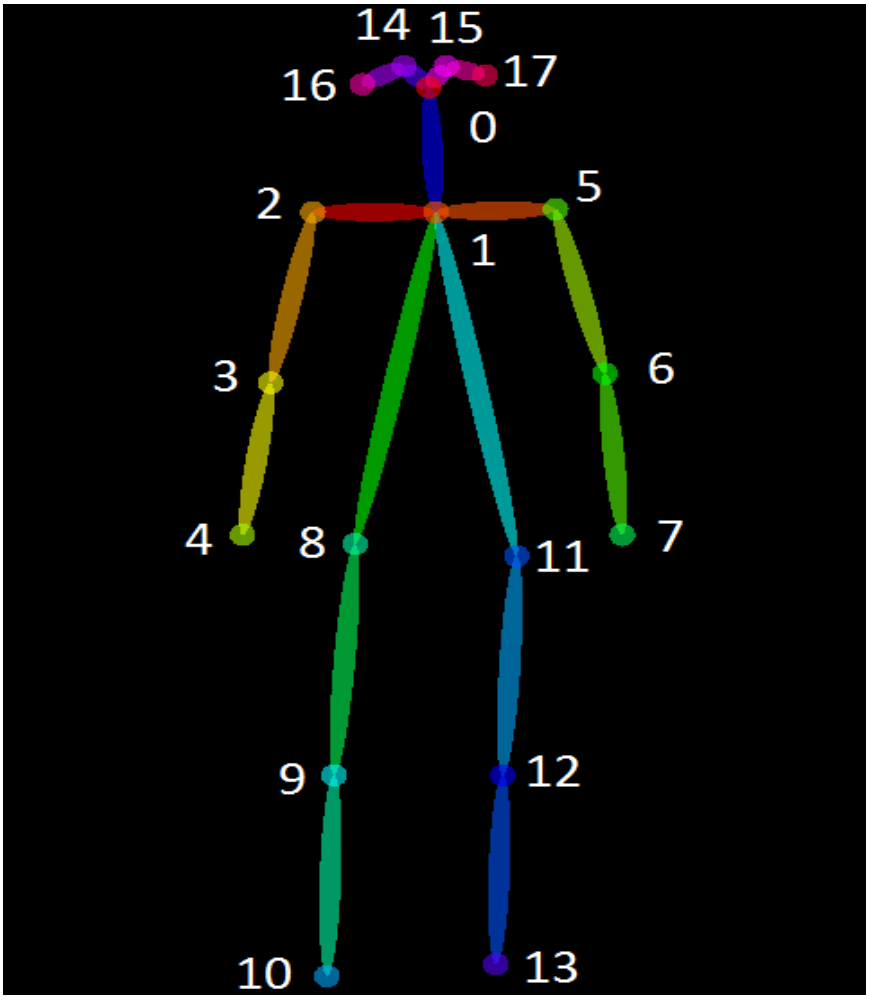


Figura 22. Estimación de los puntos y las partes anatómicas del cuerpo

Fuente: (Cao, Simon, Wei & Sheikh, 2018)

3.3.3 Localización y estimación de puntos de interés

(Güler et al., 2018), la tarea del punto clave de COCO (objetos comunes en contexto) es la detección simultanea de objetos y la localización de sus puntos claves, la importancia de la detección simultánea y estimación de punto clave es relativamente nueva se da por optar una medida novedosa inspirada en detección de objetos. Para simplificar se usa a esta tarea como detección de puntos clave y el algoritmo de predicción como el detector de punto clave.

(Guler, Neverova & Kokkinos, s.f.), la tarea del punto clave de COCO (objetos comunes en contexto) es la detección simultanea de objetos y la localización de sus puntos claves, la importancia de la detección simultánea y estimación de punto clave es relativamente nueva se da por optar una medida novedosa inspirada en detección de objetos. Para simplificar se usa a esta tarea como detección de puntos clave y el algoritmo de predicción como el detector de punto clave.

Figura 23. Detección de partes + partes asociadas de los puntos claves

Fuente: (Cao, Simon, Wei & Sheikh, 2018)

Figura 24. Detección de las articulaciones

Fuente: (Cao, Simon, Wei & Sheikh, 2018)

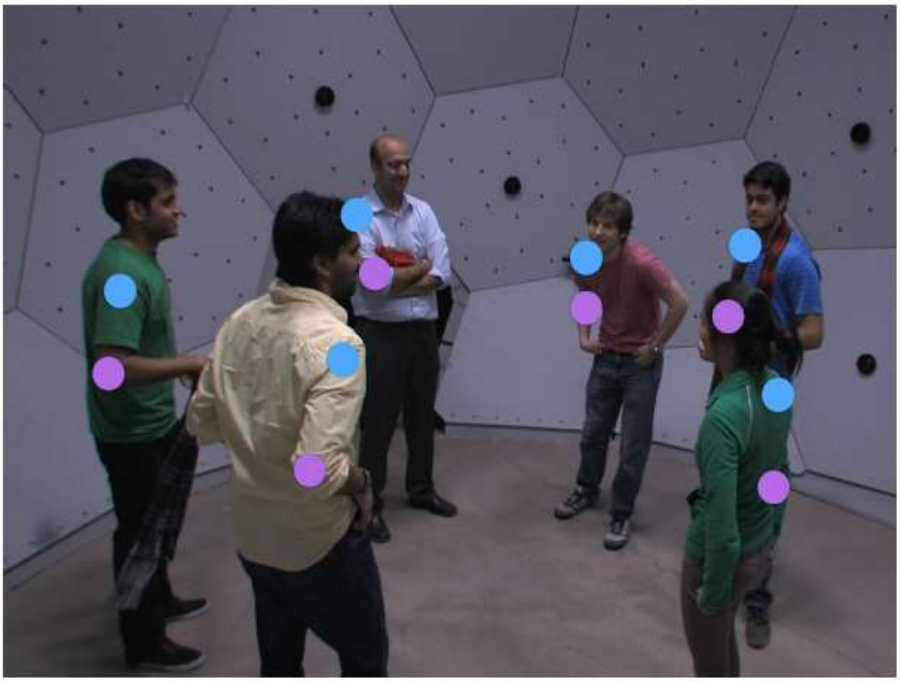


Figura 25. *Puntos clave*

Fuente: (Cao, Simon, Wei & Sheikh, 2018)

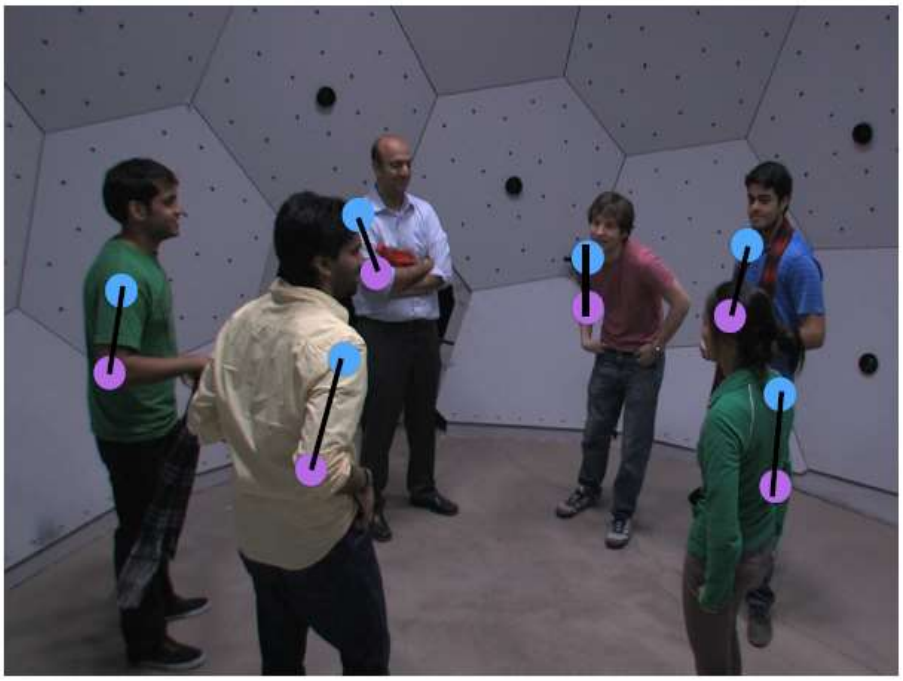


Figura 26. Partes asociadas del cuerpo

Fuente: (Cao, Simon, Wei & Sheikh, 2018)

Figura 27. Partes asociadas de las articulaciones

Fuente: (Cao, Simon, Wei & Sheikh, 2018)

Figura 28. Esqueletizacion del cuerpo

Fuente: (Cao, Simon, Wei & Sheikh, 2018)

3.4 Estimación de pose y grafos de correspondencia

En la estimación de posturas de máquinas representa los pixeles de la imagen *p-th* señal anatómica a la que se refiere como parte, $Yp \in Z \subset R2$, donde *Z* él es conjunto de todas las ubicaciones en una imagen como meta es predecir las ubicaciones de las imágenes Y= (Y1,..., YP) para todas las partes (P) de una máquina de postura (figura 26) consiste en una secuencia de predictores de clase múltiple que están entrenados para predecir la ubicación de cada parte en cada nivel de la jerarquía en cada etapa t $\in \{1. . . T\}$, donde los clasificadores g_t hace predecir de tal manera pueda asignar una ubicación a cada parte Yp = z, $\forall z \in Z$, según las características extraídas de la imagen en la ubicación Z denota por $x_z \in R^d$ información del clasificados anterior en el vecindario alrededor de cada Yp en la etapa t de un clasificador. (Cao, Simon & Wei, 2018)

Figura 29. El Campo de Afinidad permite estimar el antebrazo.

Fuente**:** (Cao, Simon, Wei & Sheikh, 2018)

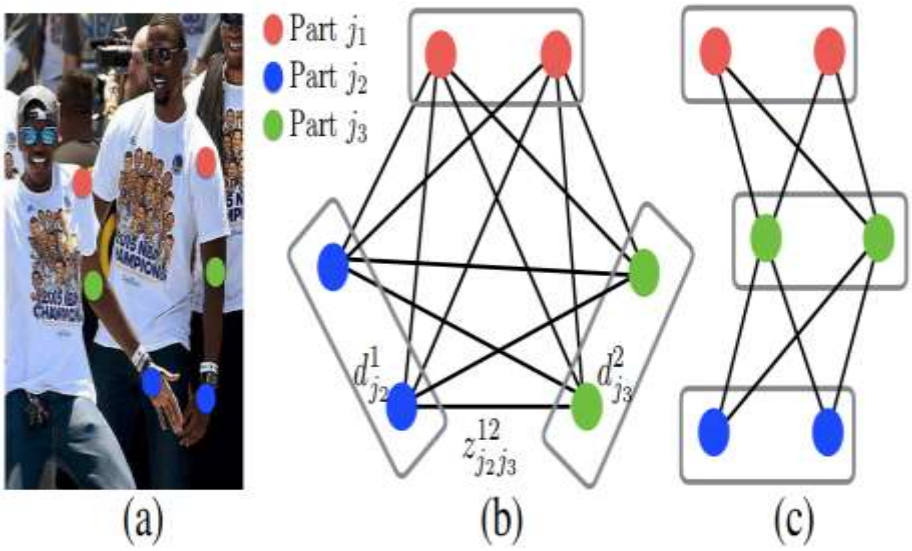


Figura 30. Grafo de correspondencia para poder articular el cuerpo

Fuente: (Cao, Simon & Wei, 2018)

A. Imagen con articulaciones detectadas. b
B. Grafo K-partes
C. Árbol de secuencia de partes.

3.5 Correspondencia e identificación de postura

Finalmente, se realiza búsqueda en base de datos de posturas para reconocer e identificar la acción que un individuo (previa detección de postura) está realizando. Algunas acciones guardadas en la base de datos son: parado, sentado, caminado, etc. Este proceso es realizado empleando una técnica de correspondencia de formas entre la postura de una persona encontrada en una imagen o video comparándola con las posturas de la base de datos.

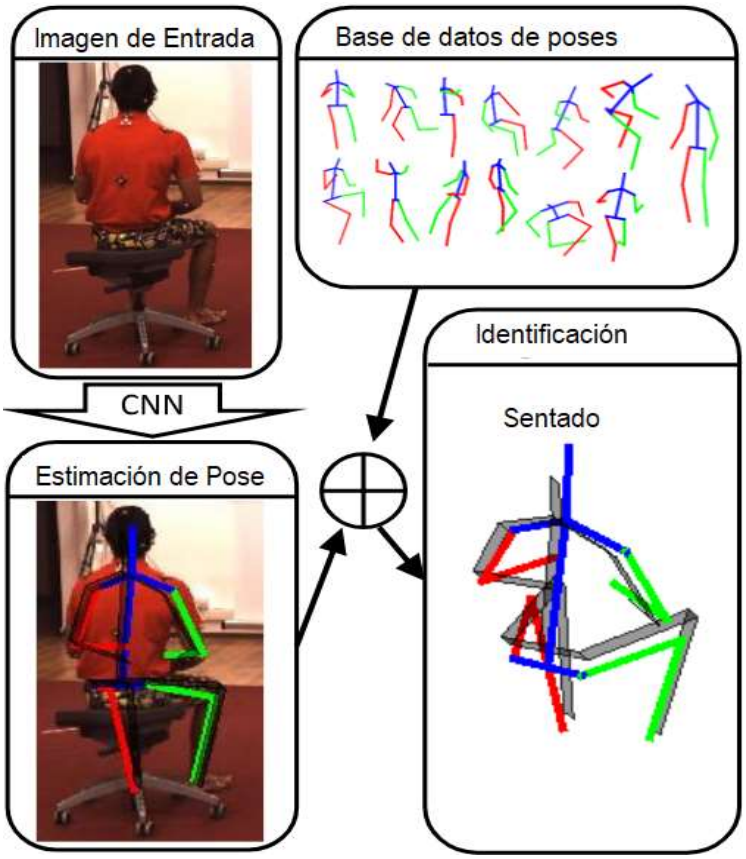


Figura 31. Técnica de correspondencia

Fuente: (Cao, Simon & Wei, 2018)

3.5.1 Distancia de frechet

(Fréchet, 1906) Es una métrica que puede ser usada para determinar cuan similares son dos curvas. Podemos interpretar este resultado como si dos entidades conectadas tuvieran que recorrer un camino, pero sin cruzarse y con la menor variación (distancia) posible.

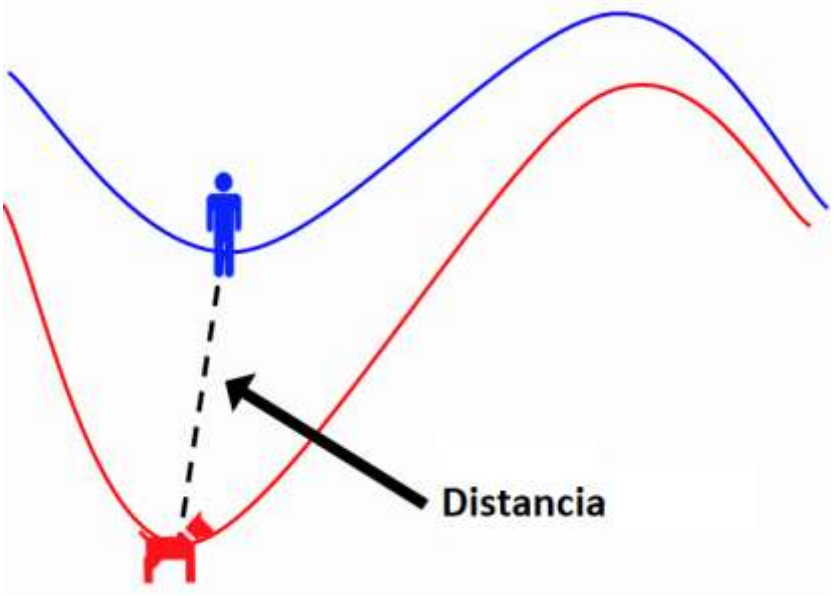


Figura 32. Analogía del cálculo de la distancia de Fréchet.

Fuentes: (Fréchet, s.f.)

Por otro lado, la correspondencia de dos curvas no es tan simple de determinar, puesto que puede tener la misma forma, pero muy distinta topología

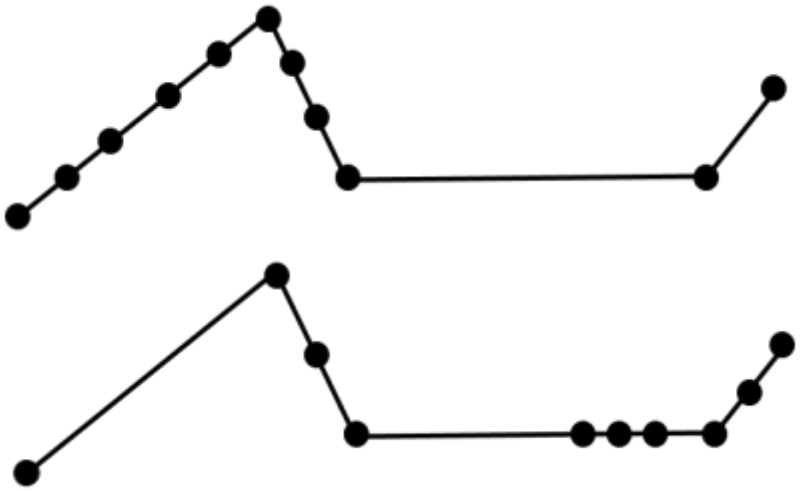


Figura 33. Curvas de forma similar pero diferente topología.

Fuentes: (Fréchet, s.f.)

lo que complica su cálculo. Entonces es necesario hacer un pre-procesamiento para que se tenga solamente los puntos o vértices necesarios.

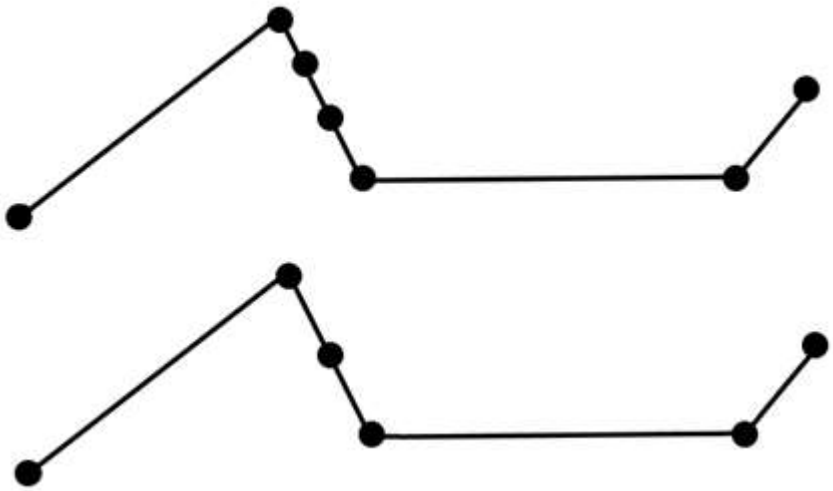


Figura 34. Simplificación topológica de curvas para mejorar el proceso de correspondencia de formas.

Fuentes: (Fréchet, 1906)

Entretanto, en nuestra propuesta tenemos las posturas siempre con la topología similar, y lo mismo almacenado en la base de datos de posturas, por lo que este proceso es mucho más simple.

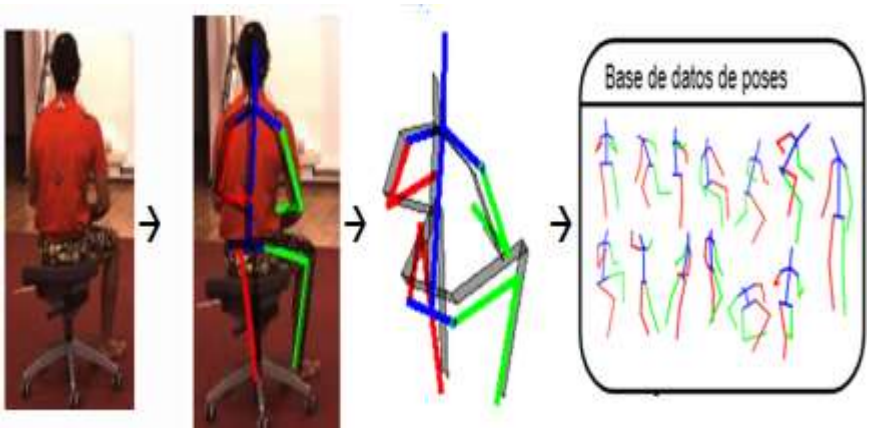


Figura 35. Proceso de correspondencia de postura.

Fuente: (Carnegie, 2016)

El algoritmo de todo el proceso de detección de esqueleto es representado el diagrama:

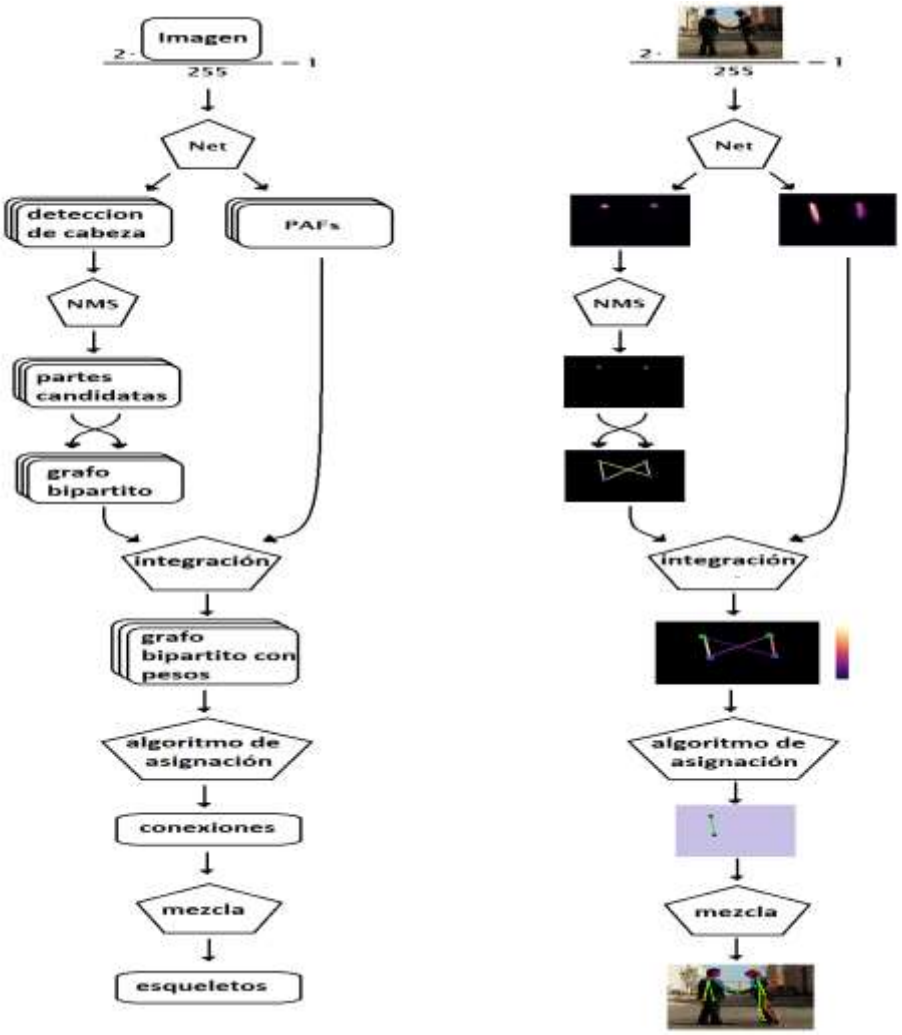


Figura 36. Algoritmo del proceso de detección

Autor: (Cao, Simon, Wei & Sheikh, 2018)

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

4.1 Diseño de la solución

4.1.1 Experimentación

El método para estimar la pose de las personas, depende del uso de una base de datos de imágenes de personas (alrededor de 100000 imágenes), de las cuales se extrajeron sus puntos clave (keypoints). Esta base de datos se evaluó y entreno en una red neuronal convolucional para obtener el conjunto de datos MPII (human multi-person dataset). Estos conjuntos de datos son colecciones de imágenes en diversos escenarios que contienen muchos aspectos reales y que son usualmente difíciles de tratar como tamaños de personas en imágenes, iluminación, opacidad, sombras, y muchos otros aspectos que son captados por cámaras.

El método utilizado es robusto y permite extraer las extremidades de las personas usando técnicas de grafos (más específicamente, estructuras de grados conectados completamente).

Algunos resultados empleando videos del centro comercial de Juliaca se muestran en las siguientes figuras.

Figura 37. Identificación de articulaciones mediante cámaras de video vigilancia

4.1.2 Base de datos

Debido a la complejidad de poder encontrar bases de datos relacionados a eventos inusuales y comportamientos en cámaras de video vigilancia se hace uso un conjunto de datasets públicos de videos en eventos y comportamientos humanos, COCO, CMU PANÓPTICO, MPII

4.1.2.1 Dataset de MPII

Es un conjunto eventos y poses humanas de MPII el conjunto de datos incluye alrededor de 25k imágenes que contienen más de 40k personas con articulaciones corporales anotadas las imágenes fueron recolectados --- sistemáticamente utilizando una taxonomía establecida de las actividades humanas diarias.

Figura 38. Base de datos de MPII

Fuente:(Max Planck Institut Informatik, s.f.)

4.1.2.2 Dataset CMU panóptico keypoint detection

El conjunto de la base de datos COCO contiene más de 200,000 imágenes y 250,000 instancias de personas con 17 puntos clave como puntos de articulaciones del cuerpo humano donde se entrenan al modelo en el conjunto de datos de COCO train2017 que incluyen imágenes de 57K e instancias y 150K personas.

Figura 39. Datasets de Panóptico CMU

Fuente: (Cao, Simon & Wei, 2018)

4.2 Resultados

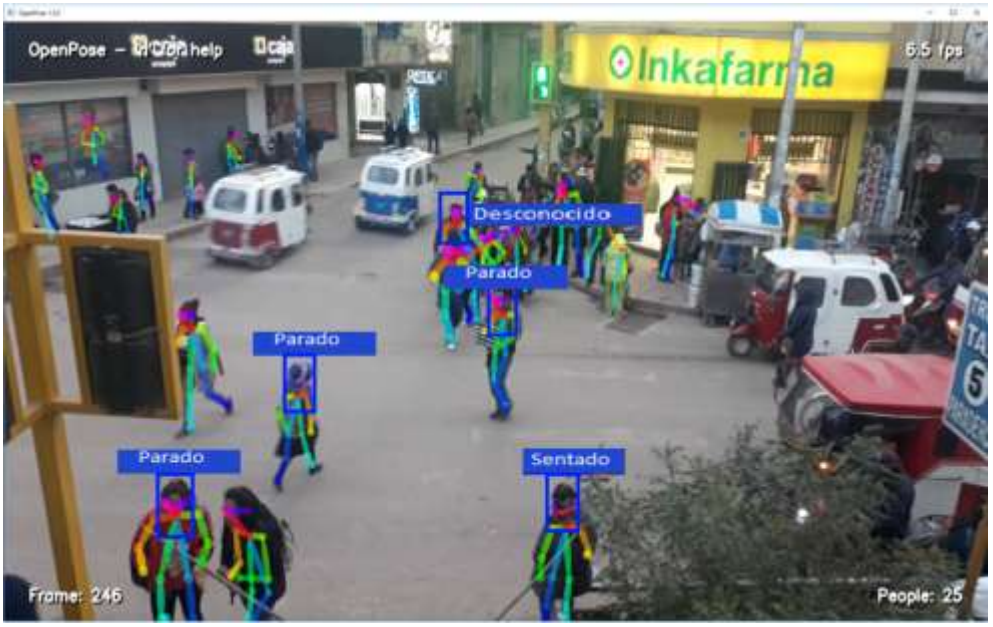


Figura 40. Identificación de eventos mediante cámaras de video vigilancia Juliaca

El sistema primeramente detecta los puntos clave en la imagen, con la técnica descrita en el capítulo 3 (máquinas de postura convolucionales), luego aplica la técnica de grafos para sí construir una estructura completa de grafos conectados (representación del esqueleto), lo cual permite estimar la pose de los individuos identificados den la imagen.

Un aspecto importante es que, en varios puntos de este documento, se menciona imagen solamente, sin embargo, no se hace distinción a un video puesto que un video es una secuencia de imágenes, por lo que cuando se trabaja con video (ya sea desde archivo o captado desde cámara) el sistema recibe secuencias de imágenes por segundo (alrededor de 30, dependiendo de la calidad del archivo de video o la capacidad de captura de la cámara), entonces se procesa la primera imagen (cuadro o frame) para estimar los puntos clave y construir el esqueleto de postura de los individuos detectados.

En el caso de videos también hay una importante consideración, a pesar de que el método es bastante rápido y eficiente una secuencia de varias imágenes por segundo, puede ser innecesario hacerlo independientemente, por lo que se hace una optimización de coherencia temporal, empleando información anterior (frames anteriores) para construir

rápidamente el esqueleto a través de un proceso de interpolación. Este proceso adicional, permite que el sistema pueda detectar las poses de las personas (esqueletos) en tiempo real.

Tabla 3

Error promedio (entre paréntesis el error estándar para cada dato)

Red Neuronal Convolucional			
Tamaño de imagen	**Entrenamiento**	**Prueba**	**Eventos**
33 X 33	0.110(0.002)	0.150(0.006)	0.020(0.006)
45 X 45	0.088(0.002)	0.156(0.008)	0.015(0.005)
70 X 70	0.075(0.002)	0.159(0.004)	0.030(0.004)
81 X 81	0.078(0.002)	0.152(0.006)	0.025(0.006)

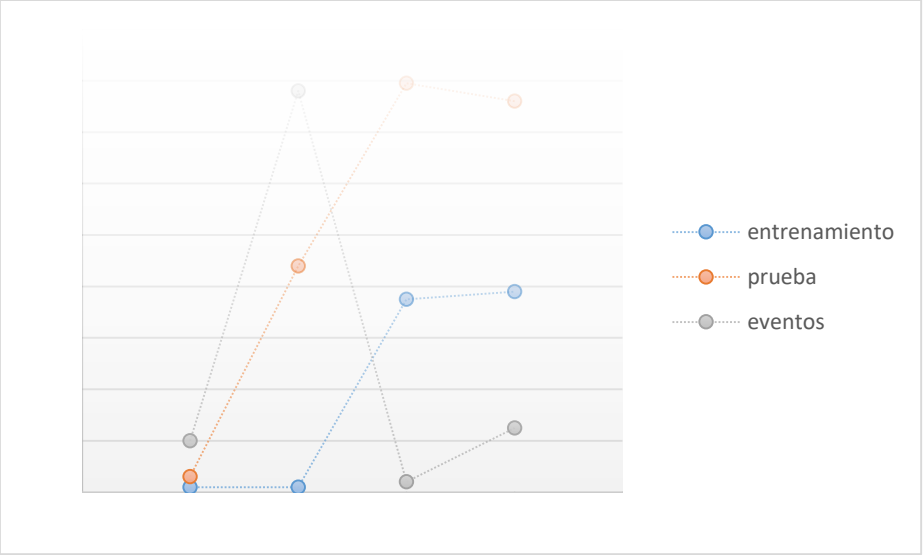


Figura 41. Gráfico de error promedio de eventos detectados

Figura 42. Detección de los puntos anatómicos mediante cámara web

Figura 43. Detección y captura de eventos mediante cámaras de video vigilancia de Juliaca

Tabla 4

Tiempo de ejecución en segundos, según el tamaño del video

Tamaño de imagen	Pre procesamiento (5 videos)	entrenamiento	Eventos
33 X 33	162.4	148.1	1.70
45 X 45	161.2	540.4	3.10
70 X 70	154.3	2260.65	6.71
81 X 81	160.6	2271.76	9.91

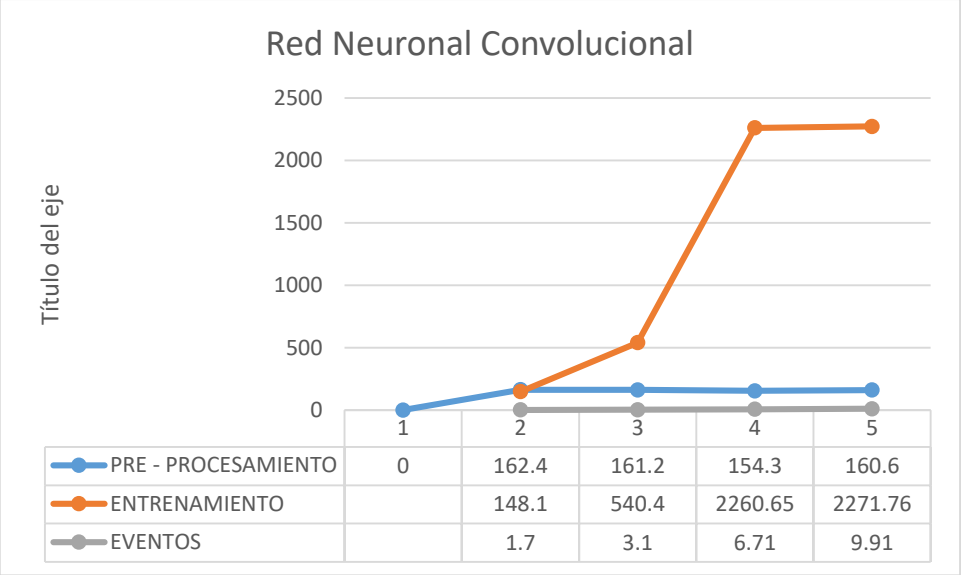


Figura 44. Grafica del tiempo de ejecución en segundos (CNN)

Tabla 5

tasa de error promedio según la pose usando la arquitectura CNN

Estimación de pose	Red neuronal convolucional	Videos propios
Perfil completo	0.120(0.015)	0.153(0.018)
Medio perfil	0.350(0.019)	0.460(0.017)
Cuarto de perfil	0.190(0.010)	0.273(0.016)
Frontal	0.030(0.004)	0.068(0.007)

El sistema es capaz de detectar a muchos individuos en un escenario. En el experimento se observa que ha detectado a la mayoría de individuos que transitan, asignando un esqueleto aproximado para cada individuo detectado, de aquí es que podemos identificar la pose de los individuos para poder estimar si corresponde a una acción normal o algo inusual.

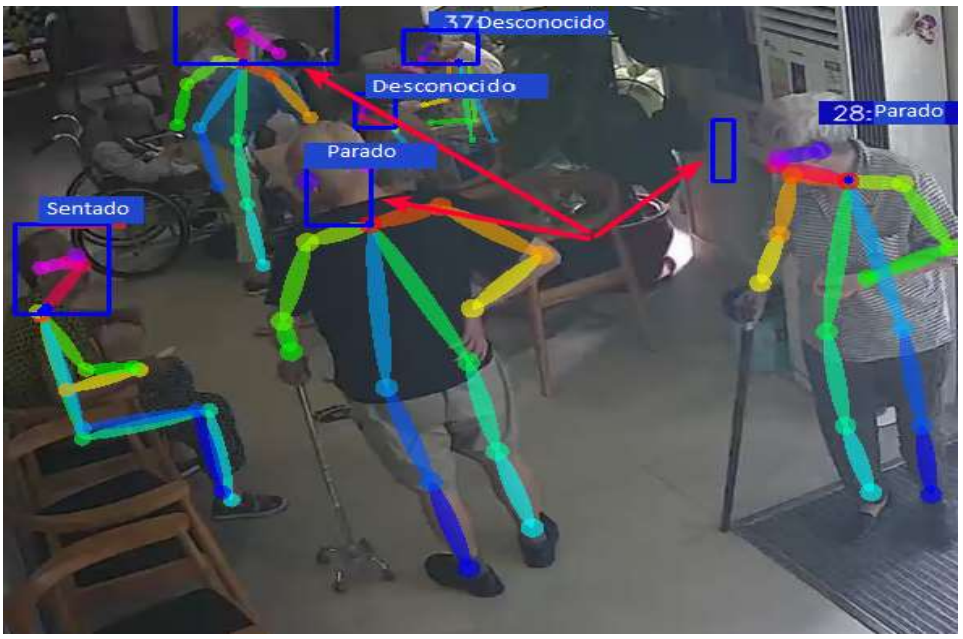
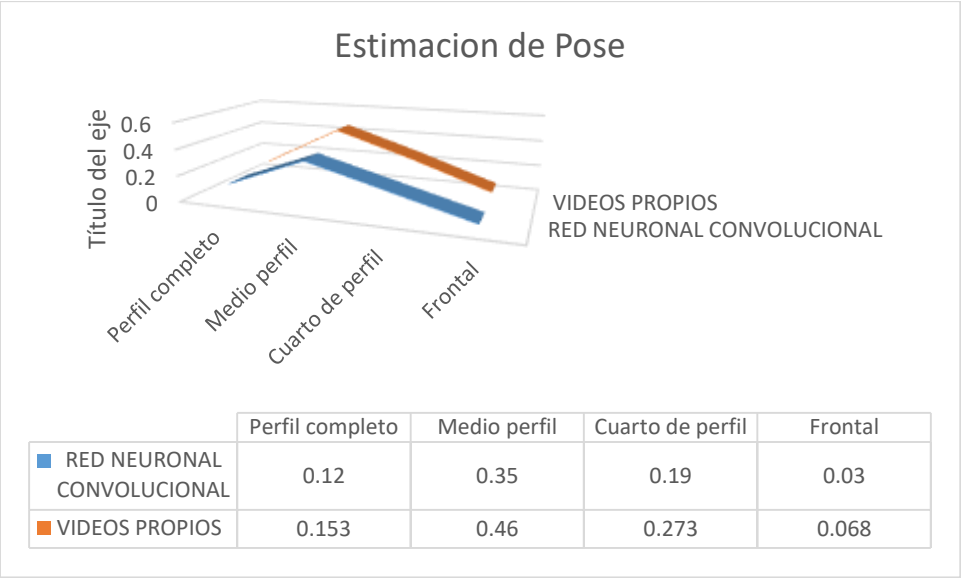


Figura 45. Identificación de los eventos detectados.



	Perfil completo	Medio perfil	Cuarto de perfil	Frontal
RED NEURONAL CONVOLUCIONAL	0.12	0.35	0.19	0.03
VIDEOS PROPIOS	0.153	0.46	0.273	0.068

Figura 46. Grafica de la estimación de Pose mediante una CNN

Una vez que tenemos los esqueletos de los individuos, se procede a usar un cálculo de correspondencia de curvas, ya que el esqueleto se considera como una curva, para poder identificar alguna acción que un individuo pueda estar realizando en un determinado momento (correspondiente a la captura de imagen).

Este proceso de correspondencia, también es basada en una búsqueda una base de datos de poses etiquetadas con acciones ya identificadas previamente, entonces lo que se hace es obtener la pose como mejor valor de correspondencia, lo que permite indicar qué está haciendo los individuos en una determinada imagen (ver figura 35).

CONCLUSIONES

- Concluimos que crear un sistema computacional capaz de procesar imágenes en video y detectar acciones y/o eventos inusuales en movimiento de personas es posible, utilizando métodos y técnicas, descritas en el presente trabajo, que están disponibles y que actualmente son viables de usar en computadores inclusive domésticos. Estas técnicas nos permiten poder detectar o capturar los movimientos de personas y analizar la postura y de ese modo determinar características o comportamientos inusuales. Las redes convolucionales, actualmente, permiten una extracción de puntos óptimos que corresponden a las principales articulaciones de una persona (hombros, cuello, rodilla, codos, etc.). Otra de las técnicas que se han utilizado es el uso de los campos de afinidad de partes y las técnicas de los grafos de correspondencia donde se hace uso de "deep learning" con redes neuronales convolucionales. Esto permite mejorar drásticamente la capacidad de extracción de características en los video analizados.
- Finalmente, para el uso del sistema propuesto en una cámara de video vigilancia, esta debe estar conectada a un computador de alta capacidad para emplear las técnicas descritas, de manera que permita mostrar automáticamente todos los posibles eventos identificados correspondientes a los movimientos de personas en las escenas captadas por las camaras.

RECOMENDACIONES

- Debido a la inseguridad que se vive actualmente, la detección de eventos inusuales es uno de los temas de investigación más importantes en la actualidad, ya que se quiere alertar de manera rápida posibles situaciones peligrosas y evitar finales lamentables. Por lo tanto, se recomienda profundizar en temas relacionados a la detección de rasgos y características en imágenes, con el fin de analizar todo tipo de actividad fuera de lo normal. Este proceso debe, necesariamente, ser automatizado puedo que la cantidad de cámaras crece día a día, y sería imposible hacerlo de manera manual o individual.

- Como trabajo futuro, queremos mejorar el proceso de identificación de evento inusual complementándolo con un proceso de seguimiento (tracking), ya que existen muchas posibilidad para indicar un evento inusual cuando no lo es realmente, entonces un sistema que sea realmente útil debe tener un alto grado de seguridad en reportar un evento inusual y evitar falsos positivos, esto solamente es posible lograr si es que se hace un seguimiento a los movimientos de un individuo por un periodo de tiempo para garantizar que se trata realmente de un evento inusual.

- El trabajo también puede ser extendido para ser aplicado en otro tipo de objetos con movimientos como vehículos y animales.

BIBLIOGRAFÍA

Amador, E., & Baumela, L. (2017). *Estimación Precisa de la Orientación del Rostro Humano Utilizando Redes de Neuronas*. Retrieved from http://oa.upm.es/47220/1/TFG_ELVIRA_AMADOR_DOMINGUEZ.pdf

Artificial Intelligence and Machine Learning for Dummies. (2018). Retrieved February 2, 2019, from https://www.theninjacto.xyz/Artificial-Intelligence-and-Machine-Learning-for-Dummies/

Barbuzza, R., Fernández, L., Domínguez, L., Pérez, A., Rubiales, A., D'Amato, J. (2017). Separación de sombras a los objetos detectados con sustracción de fondo en video. *Digital.Cic.Gba.Gob.Ar*. Retrieved from https://digital.cic.gba.gob.ar/handle/11746/6543

Bulat, A., & Tzimiropoulos, G. (2017). *Human pose estimation via Convolutional Part Heatmap Regression*. Retrieved from http://www.cs.nott.ac.uk/~psxab5/

Calvo (2017). Red Neuronal Recurrente - RNN - Diego Calvo. Retrieved April 26, 2019, from http://www.diegocalvo.es/red-neuronal-recurrente/

Cao, Z., Simon, T., Wei, S., & Sheikh (2017). *Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields*. Retrieved from http://openaccess.thecvf.com/content_cvpr_2017/papers/Cao_Realtime_Multi-Person_2D_CVPR_2017_paper.pdf

Cobos & González (2015). *Localización y seguimiento de personas en entornos Videovigilados*. Retrieved from https://e-archivo.uc3m.es/bitstream/handle/10016/26069/PFC_Luis_MartinCobos_Blanco.pdf?sequence=1&isAllowed=y

Dalal, N., & Triggs B.(2005). Histogramas de gradientes orientados para la detección humana. *Ieeexplore.Ieee.Org*. Retrieved from http://ieeexplore.ieee.org/abstract/document/1467360/

Eichner, M., & Ferrari, V. (2012). Human Pose Co-Estimation and Applications. *IEEE*

Transactions on Pattern Analysis and Machine Intelligence, *34*(11), 2282–2288. https://doi.org/10.1109/TPAMI.2012.85

Ferrari, V., Marin, M., & Zisserman, A. (2008). Progressive search space reduction for human pose estimation. *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 1–8. https://doi.org/10.1109/CVPR.2008.4587468

Fréchet, M. (s.f.). *Sur quelques points du calcul fonctionnel, par M. Maurice Fréchet, ...* Retrieved from https://www.worldcat.org/title/sur-quelques-points-du-calcul-fonctionnel-par-m-maurice-frechet/oclc/457373324

Fuentes, M., & Velastin, S. (2004). Vigilancia avanzada: del tracking a la deteccion de sucesos. *Ieeexplore.Ieee.Org*. Retrieved from https://www.semanticscholar.org/paper/Vigilancia-Avanzada%3A-del-tracking-a-la-detecci%C3%B3n-de-Fuentes-Velastin/462d3651c2b4a52789466419faea60f6d8088154

Gervasoni, L., Damato, J., Barbuzza, R., & Vénere, M. (2014). *Un metodo eficiente para la sustraccion de fondo en videos usando gpu*. Retrieved from http://ri.conicet.gov.ar/handle/11336/9413

Guevara, M., Echeverry, J., & Ardila W. (2008). Scientia et technica. In *Scientia et technica* (Vol. 1). Retrieved from http://revistas.utp.edu.co/index.php/revistaciencia/article/view/3679/2069

Güler, R., Neverova, N., & Kokkinos, I. (2018). *DensePose: Dense Human Pose Estimation In The Wild*. Retrieved from https://arxiv.org/abs/1802.00434

Hernández, A., Reyes, M., Ponce, V., & Escalera, S. (2012). GrabCut-Based Human Segmentation in Video Sequences. *Sensors*, *12*(11), 15376–15393. https://doi.org/10.3390/s121115376

Hernández, R., García E., Ramos, J., & Guil, N. (2014). Modelos de representación de características para la clasificación de acciones humanas en video: estado del arte. *Revista Cubana de Ciencias Informáticas*, *8*(4), 21–51. Retrieved from http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992014000400002&lng=es&nrm=iso&tlng=pt

Krizhevsky, A., Sutskever, I., & Hinton, G. (2012). *ImageNet Classification with Deep Convolutional Neural Networks* Retrieved from http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networ

Ledezma (2012). *Maquinas De Soporte Vectorial (svm) Para La Detección De Nódulos*

Pulmonares En Tomografía Axial Computarizada (TAC). (Pontificia Universidad Javeriana Facultad de Ingeniería Maestría en Ingeniería Electrónica Bogotá). Retrieved from https://repository.javeriana.edu.co/bitstream/handle/10554/12741/LedezmaGarridoWillmar2012.pdf?sequence=1&isAllowed=y

Letelier, P. (2006). *Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. Retrieved from http://www.cyta.com.ar/ta0502/b_v5n2a1.htm

Manejes & Camara. (2017). *Deteccion de eventos Anomalos en Videos*. Retrieved from http://repositorio.ucsp.edu.pe/bitstream/UCSP/15400/1/MENEJES_PALOMINO_NEP_DET.pdf

Martínez, L. (2018). Identificación automática de acciones humanas en secuencias de video para soporte de videovigilancia. *Tesis.Pucp.Edu.Pe*. Retrieved from http://tesis.pucp.edu.pe/repositorio/handle/123456789/13049

Max planck institut informatik. (s.f.). Base de datos de posturas humanas MPII. Retrieved June 25, 2019, from https://www.mpi-inf.mpg.de/institute/

Nakada, M., Wang, H., & Terzopoulos, D. (2017). AcFR: Active Face Recognition Using Convolutional Neural Networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 35–40. https://doi.org/10.1109/CVPRW.2017.11

Osorio, E. & Holguín, G. (2015). *Deteccion Automatizada de Objetos en Secuencias de Video Utilizando HOG*. Retrieved from https://core.ac.uk/download/pdf/71398633.pdf

Otero, R., (2012). *Reconocimiento de localizaciones mediante máquinas de soporte vectorial*. Retrieved from https://e-archivo.uc3m.es/handle/10016/15319

Pantrigo, J. (s.f.). *Algoritmos De Optimización Aplicados Al Seguimiento Del Movimiento Articular Y La Digitalización Automática Del Movimiento Humano*. Retrieved from https://www.academia.edu/20790426/Algoritmos_De_Optimizaci%C3%93n_Aplicados_Al_Seguimiento_Del_Movimiento_Articular_Y_La_Digitalizaci%C3%93n_Autom%C3%81tica_Del_Movimiento_Humano

Ramírez, M., Travieso, C.G., Calderon, A., Hernandez, J., Salas, O., Mora, F., Prendas, J.P. (2013). Detección y seguimiento de objetos presentes en video 2D con MatLab. *UNICIENCIA*, *27*(2), 39–50. Retrieved from https://www.redalyc.org/pdf/4759/475947763004.pdf

Reina, R., & Real, P., (2011). *Esqueletización de imágenes 3D*. Retrieved from http://master.us.es/masterma1/TfM_pdfs(Julio11)/Raul_Reina_Molina.pdf

Realpe, M., Vintimilla, B., Romero, D., & Remagnino, P. (s.f.). *Análisis de comportamiento humano: Metodología para localización y seguimiento de personas en secuencias de video*. Retrieved from http://www.iiis.org/CDs2009/CD2009CSC/CISCI2009/PapersPdf/C629DZ.pdf

Sanlam (2018). Retrieved from https://www.sanlamgis.com/Documents/SGIP - Introduccion a la Inteligencia Artificial y el Aprendizaje de Máquina.pdf

Sidenbladh, H. (2004). Detecting human motion with support vector machines. *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, 188-191 Vol.2. https://doi.org/10.1109/ICPR.2004.1334092

Simon, T., Joo, H., Matthews, I., & Sheikh (2017). Hand Keypoint Detection in Single Images Using Multiview Bootstrapping. *Openaccess.Thecvf.Com*. Retrieved from http://openaccess.thecvf.com/content_cvpr_2017/papers/Simon_Hand_Keypoint_Detection_CVPR_2017_paper.pdf

Vizcaya, R., Flores, J., & Lazcano S. (2017). *Desempeño de una red neuronal convolucional para clasificación de señales de tránsito*. Retrieved from https://www.researchgate.net/publication/323456954

Wei, S., Ramakrishna, V., Kanade, T., & Sheikh, Y. (2016). *Convolutional Pose Machines*. Retrieved from https://www.cv-foundation.org/openaccess/content_cvpr_2016/app/S20-08.pdf

ANEXOS

Anexo 1. Matriz de consistencia

Problema	Objetivos	Indicadores	Metodología
Dada la inseguridad en la que actualmente vivimos se ha hecho necesario buscar alternativas para poder disminuir estos acontecimientos. De hecho, no es simple hablar de disminuir o evitar la violencia que se vive en el país, sin embargo, podemos intentar enfrentar este problema aprovechando la disponibilidad de cámaras de video vigilancia que ha ido incrementando cada vez más, entonces es posible utilizar estos recursos para poder analizar imágenes (video) e interpretar posibles acciones de violencia, sospechosas o dudosas.	Implementar una aplicación computacional para "detección de eventos inusuales en imágenes y video de cámaras de vigilancia" **Objetivos específicos** - Analizar y procesar imágenes y video. - Detectar patrones, imágenes y video - Automatizar y alertar automáticamente	- Imágenes - videos - Patrones - Aplicación	- Metodología ágil XP "programación extrema" - Arquitectura CNN -

¿Mejoraría la seguridad si se analiza las imágenes (video) captadas por las cámaras de video?			

Anexo 2. Código Fuente

```
enum class PoseModel : unsigned char
{
 /**
  * COCO + 6 foot keypoints + neck + lower abs model, with 25+1 components .
  */
 BODY_25 = 0,
 COCO_18,              //< COCO model + neck, with 18+1 components (see poseParameters.hpp for details).
 MPI_15,               //< MPI model, with 15+1 components (see poseParameters.hpp for details).
 MPI_15_4,             //< Variation of the MPI model, reduced number of CNN stages to 4
 Size,
};

enum class PoseProperty : unsigned char
{
 NMSThreshold = 0,
 ConnectInterMinAboveThreshold,
 ConnectInterThreshold,
 ConnectMinSubsetCnt,
 ConnectMinSubsetScore,
 Size,
};
Array<float> PoseExtractor::getPoseKeypoints() const
{
 try
 {
  return spPoseExtractorNet->getPoseKeypoints();
 }
 catch (const std::exception& e)
 {
  error(e.what(), __LINE__, __FUNCTION__, __FILE__);
  return Array<float>{};
 }
}
PoseExtractorNet::PoseExtractorNet(const PoseModel poseModel, const std::vector<HeatMapType>& heatMapTypes, const ScaleMode heatMapScaleMode, const bool addPartCandidates, const bool maximizePositives) :
 mPoseModel{poseModel},
 mNetOutputSize{0,0},
 mHeatMapTypes{heatMapTypes},
 mHeatMapScaleMode{heatMapScaleMode},
 mAddPartCandidates{addPartCandidates}
{
 try
 {
```

```
    // Error check
    if (mHeatMapScaleMode != ScaleMode::ZeroToOne && mHeatMapScaleMode !=
ScaleMode::PlusMinusOne
      && mHeatMapScaleMode != ScaleMode::UnsignedChar && mHeatMapScaleMode
!= ScaleMode::NoScale)
      error("The ScaleMode heatMapScaleMode must be ZeroToOne, PlusMinusOne,
UnsignedChar, or NoScale.",
          __LINE__, __FUNCTION__, __FILE__);
    // Properties - Init to 0
    for (auto& property : mProperties)
      property = 0.;
    // Properties - Fill default values
    mProperties[(int)PoseProperty::NMSThreshold]                              =
getPoseDefaultNmsThreshold(mPoseModel, maximizePositives);
    mProperties[(int)PoseProperty::ConnectInterMinAboveThreshold]
      = getPoseDefaultConnectInterMinAboveThreshold(maximizePositives);
    mProperties[(int)PoseProperty::ConnectInterThreshold]                     =
getPoseDefaultConnectInterThreshold(
      mPoseModel, maximizePositives);
    mProperties[(int)PoseProperty::ConnectMinSubsetCnt]                       =
getPoseDefaultMinSubsetCnt(maximizePositives);
    mProperties[(int)PoseProperty::ConnectMinSubsetScore]                     =
getPoseDefaultConnectMinSubsetScore(
      maximizePositives);
  }
  catch (const std::exception& e)
  {
    error(e.what(), __LINE__, __FUNCTION__, __FILE__);
  }
}
inline Rectangle<float> getHandFromPoseIndexes(const Array<float>& poseKeypoints,
const unsigned int person, const unsigned int wrist,
                const unsigned int elbow, const unsigned int shoulder, const float
threshold)
{
  try
  {
    Rectangle<float> handRectangle;
    // Parameters
    const                auto*                posePtr                =
&poseKeypoints.at(person*poseKeypoints.getSize(1)*poseKeypoints.getSize(2));
    const auto wristScoreAbove = (posePtr[wrist*3+2] > threshold);
    const auto elbowScoreAbove = (posePtr[elbow*3+2] > threshold);
    const auto shoulderScoreAbove = (posePtr[shoulder*3+2] > threshold);
    const auto ratioWristElbow = 0.33f;
    // Hand
    if (wristScoreAbove && elbowScoreAbove && shoulderScoreAbove)
    {
```

```
    // pos_hand = pos_wrist + ratio * (pos_wrist - pos_elbox) = (1 + ratio) * pos_wrist - ratio * pos_elbox
    handRectangle.x = posePtr[wrist*3] + ratioWristElbow * (posePtr[wrist*3] - posePtr[elbow*3]);
    handRectangle.y = posePtr[wrist*3+1] + ratioWristElbow * (posePtr[wrist*3+1] - posePtr[elbow*3+1]);
    const auto distanceWristElbow = getDistance(poseKeypoints, person, wrist, elbow);
    const auto distanceElbowShoulder = getDistance(poseKeypoints, person, elbow, shoulder);
    handRectangle.width = 1.5f * fastMax(distanceWristElbow, 0.9f * distanceElbowShoulder);
   }
   // height = width
   handRectangle.height = handRectangle.width;
   // x-y refers to the center --> offset to topLeft point
   handRectangle.x -= handRectangle.width / 2.f;
   handRectangle.y -= handRectangle.height / 2.f;
   // Return result
   return handRectangle;
  }
  catch (const std::exception& e)
  {
   error(e.what(), __LINE__, __FUNCTION__, __FILE__);
   return Rectangle<float>{};
  }
}

inline std::array<Rectangle<float>, 2> getHandFromPoseIndexes(const Array<float>& poseKeypoints, const unsigned int person,
                                                              const unsigned int lWrist, const unsigned int lElbow, const unsigned int lShoulder,
                                                              const unsigned int rWrist, const unsigned int rElbow, const unsigned int rShoulder,
                                                              const float threshold)
{
  try
  {
   return {getHandFromPoseIndexes(poseKeypoints, person, lWrist, lElbow, lShoulder, threshold),
       getHandFromPoseIndexes(poseKeypoints, person, rWrist, rElbow, rShoulder, threshold)};
  }
  catch (const std::exception& e)
  {
   error(e.what(), __LINE__, __FUNCTION__, __FILE__);
   return std::array<Rectangle<float>, 2>(); // Parentheses instead of braces to avoid error in GCC 4.8
  }
}
```

```
float getAreaRatio(const Rectangle<float>& rectangleA, const Rectangle<float>& rectangleB)
{
 try
 {
  // https://stackoverflow.com/a/22613463
  const auto sA = rectangleA.area();
  const auto sB = rectangleB.area();
  const auto bottomRightA = rectangleA.bottomRight();
  const auto bottomRightB = rectangleB.bottomRight();
  const auto sI = fastMax(0.f, 1.f + fastMin(bottomRightA.x, bottomRightB.x) - fastMax(rectangleA.x, rectangleB.x))
      * fastMax(0.f, 1.f + fastMin(bottomRightA.y, bottomRightB.y) - fastMax(rectangleA.y, rectangleB.y));
  // // Option a - areaRatio = 1.f only if both Rectangle has same size and location
  // const auto sU = sA + sB - sI;
  // return sI / (float)sU;
  // Option b - areaRatio = 1.f if at least one Rectangle is contained in the other
  const auto sU = fastMin(sA, sB);
  return fastMin(1.f, sI / (float)sU);
 }
 catch (const std::exception& e)
 {
  error(e.what(), __LINE__, __FUNCTION__, __FILE__);
  return 0.f;
 }
}

void trackHand(Rectangle<float>& currentRectangle, const std::vector<Rectangle<float>>& previousHands)
{
 try
 {
  if (currentRectangle.area() > 0 && previousHands.size() > 0)
  {
   // Find closest previous rectangle
   auto maxIndex = -1;
   auto maxValue = 0.f;
   for (auto previous = 0u ; previous < previousHands.size() ; previous++)
   {
    const auto areaRatio = getAreaRatio(currentRectangle, previousHands[previous]);
    if (maxValue < areaRatio)
    {
     maxValue = areaRatio;
     maxIndex = previous;
    }
   }
   // Update current rectangle with closest previous rectangle
```

```
    if (maxIndex > -1)
    {
      const auto& prevRectangle = previousHands[maxIndex];
      const auto ratio = 2.f;
      const auto newWidth = fastMax((currentRectangle.width * ratio + prevRectangle.width) * 0.5f,
                    (currentRectangle.height * ratio + prevRectangle.height) * 0.5f);
      currentRectangle.x = 0.5f * (currentRectangle.x + prevRectangle.x + 0.5f * (currentRectangle.width + prevRectangle.width) - newWidth);
      currentRectangle.y = 0.5f * (currentRectangle.y + prevRectangle.y + 0.5f * (currentRectangle.height + prevRectangle.height) - newWidth);
      currentRectangle.width = newWidth;
      currentRectangle.height = newWidth;
    }
   }
  }
  catch (const std::exception& e)
  {
   error(e.what(), __LINE__, __FUNCTION__, __FILE__);
  }
}

HandDetector::HandDetector(const PoseModel poseModel) :  mPoseIndexes(
getPoseKeypoints(poseModel, {"LWrist", "LElbow", "LShoulder", "RWrist", "RElbow", "RShoulder"})),
  mCurrentId{0}
{
}

HandDetector::~HandDetector()
{
}

std::vector<std::array<Rectangle<float>, 2>> HandDetector::detectHands(const Array<float>& poseKeypoints) const
{
  try
  {
   const auto numberPeople = poseKeypoints.getSize(0);
   std::vector<std::array<Rectangle<float>, 2>> handRectangles(numberPeople);
   const auto threshold = 0.03f;
   // If no poseKeypoints detected -> no way to detect hand location
   // Otherwise, get hand position(s)
   if (!poseKeypoints.empty())
   {
    for (auto person = 0 ; person < numberPeople ; person++)
    {
     handRectangles.at(person) = getHandFromPoseIndexes(
```

```
        poseKeypoints, person, mPoseIndexes[(int)PosePart::LWrist],
mPoseIndexes[(int)PosePart::LElbow],
        mPoseIndexes[(int)PosePart::LShoulder], mPoseIndexes[(int)PosePart::RWrist],
        mPoseIndexes[(int)PosePart::RElbow], mPoseIndexes[(int)PosePart::RShoulder],
threshold
       );
      }
    }
    return handRectangles;
  }
  catch (const std::exception& e)
  {
    error(e.what(), __LINE__, __FUNCTION__, __FILE__);
    return std::vector<std::array<Rectangle<float>, 2>>{};
  }
}

std::vector<std::array<Rectangle<float>, 2>> HandDetector::trackHands(const
Array<float>& poseKeypoints)
{
  try
  {
    std::lock_guard<std::mutex> lock{mMutex};
    // Baseline detectHands
    auto handRectangles = detectHands(poseKeypoints);
    // If previous hands saved
    for (auto& handRectangle : handRectangles)
    {
      trackHand(handRectangle[0], mHandLeftPrevious);
      trackHand(handRectangle[1], mHandRightPrevious);
    }
    // Return result
    return handRectangles;
  }
  catch (const std::exception& e)
  {
    error(e.what(), __LINE__, __FUNCTION__, __FILE__);
    return std::vector<std::array<Rectangle<float>, 2>>{};
  }
}

void HandDetector::updateTracker(const std::array<Array<float>, 2>& handKeypoints,
const unsigned long long id)
{
  try
  {
    std::lock_guard<std::mutex> lock{mMutex};
    if (mCurrentId < id)
    {
```

```
    mCurrentId = id;
    // Parameters
    const auto numberPeople = handKeypoints.at(0).getSize(0);
    const auto thresholdRectangle = 0.25f;
    // Update pose keypoints and hand rectangles
    mPoseTrack.resize(numberPeople);
    mHandLeftPrevious.clear();
    mHandRightPrevious.clear();
    for (auto person = 0u ; person < mPoseTrack.size() ; person++)
    {
     const auto scoreThreshold = 0.66667f;
     // Left hand
     if (getAverageScore(handKeypoints[0], person) > scoreThreshold)
     {
      const auto handLeftRectangle = getKeypointsRectangle(handKeypoints[0],
person, thresholdRectangle);
      if (handLeftRectangle.area() > 0)
       mHandLeftPrevious.emplace_back(handLeftRectangle);
     }
     // Right hand
     if (getAverageScore(handKeypoints[1], person) > scoreThreshold)
     {
      const auto handRightRectangle = getKeypointsRectangle(handKeypoints[1],
person, thresholdRectangle);
      if (handRightRectangle.area() > 0)
       mHandRightPrevious.emplace_back(handRightRectangle);
     }
    }
   }
  }
  catch (const std::exception& e)
  {
   error(e.what(), __LINE__, __FUNCTION__, __FILE__);
  }
}

std::array<unsigned int, (int)HandDetector::PosePart::Size>
HandDetector::getPoseKeypoints(
  const PoseModel poseModel, const std::array<std::string,
(int)HandDetector::PosePart::Size>& poseStrings
) const
{
  std::array<unsigned int, (int)PosePart::Size> poseKeypoints;
  for (auto i = 0u ; i < poseKeypoints.size() ; i++)
   poseKeypoints.at(i) = poseBodyPartMapStringToKey(poseModel, poseStrings.at(i));
  return poseKeypoints;
                                                  }
```