UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA

MAESTRÍA EN INGENIERÍA ELÉCTRICA - PROCESAMIENTO DIGITAL DE SEÑALES

ANÁLISIS Y PROCESAMIENTO DE VÍDEO PARA LA DETECCIÓN DE VÍCTIMAS EN ENTORNOS POST-DESASTRE

TESIS

QUE PARA OPTAR POR EL GRADO DE:

MAESTRO EN INGENIERÍA

PRESENTA:

CARLOS IGNACIO GARCÍA SÁNCHEZ

TUTOR

M.I. LARRY HIPÓLITO ESCOBAR SALGUERO

FACULTAD DE INGENIERÍA

CIUDAD UNIVERSITARIA, CDMX AGOSTO 2019

JURADO ASIGNADO:

Presidente:	Dr. Gastélum Strozzi Alfonso
Secretario:	Dr. Rivera Rivera Carlos
Primer vocal:	M.I. Escobar Salguero Larry Hipólito
Segundo vocal:	M.I. Minami Koyama Yukihiro
Tercer vocal:	Dr. Rascón Estebané Caleb

Lugar de realización de la tesis:
Laboratorio de Procesamiento Digital de Señales, segundo piso Edificio T
Posgrado de Ingeniería UNAM

TUTOR DE TESIS:

M.I. LARRY HIPÓLITO ESCOBAR SALGUERO

.....

FIRMA

Le dedicó este trabajo a mis padres, familiares y seres queridos que me apoyaron durante mis estudios y desarrollo de este proyecto.

Resumen

En este trabajo se presenta el planteamiento e implementación de un sistema de procesamiento de imágenes, desarrollado para detectar personas ocluidas parcialmente, con la finalidad de auxiliar la búsqueda de víctimas en entornos post-desastre como en edificaciones parcialmente colapsadas después de un temblor.

El sistema propuesto adquiere imágenes de la transmisión de vídeo de dos cámaras: una sensible al intervalo del espectro electromagnético visible por el ojo humano, denotada VIS en inglés y la segunda a la banda infrarroja de longitud de onda larga de siglas LWIR, relacionada con la radiación térmica. Ambos dispositivos se instalaron en una base para lograr un *campo de visión* compartido y establecer una regla de correspondencia de puntos entre las dos imágenes a nivel de pixeles, utilizando geometría de proyección.

La disposición de cámaras descrita se planteó para agilizar y resolver la detección de personas a pesar de las condiciones de la aplicación, atendiendo particularmente la *difusa distinción* del entorno de los objetivos, al buscar y determinar las regiones de análisis mediante vestigios de calor y la función de correspondencia entre las imágenes.

También se examinó la contribución que las imágenes térmicas pueden hacer a un clasificador, implementado cinco modelos fundamentados en dos arquitecturas de redes neuronales, con los que se experimentó entrenamientos de reajuste y completos siguiendo la metodología de observaciones locales formulada en *redes neuronales convolucionales regionales*, para afrontar la visibilidad parcial de los objetivos en conjunto con el uso de *descriptores de características*. Este proceso de aprendizaje se desarrolló con una colección de imágenes VIS y pares LWIR-VIS de personas y víctimas simuladas.

Para desarrollar y probar la regla de correlación de pixeles entre las dos cámaras, se elaboró un *patrón de calibración* perceptible para ambas bandas del espectro, y se consiguió establecer la relación de correspondencia con un error menor a un pixel y una rectificación estereoscópica parcial del 75 % de la resolución.

Y la evaluación de los clasificadores expuso la baja aportación de texturas de las imágenes LWIR al examinar por separado las fuentes, pero al utilizar un espacio menos dependiente de

descripciones locales como los mapas de bordes, las proyecciones térmicas demostraron favorecer a los clasificadores y a la etapa de detección, indicando la factibilidad de desarrollar un modelo que analice la imagen rectificada de cuatro canales, formada por los cuadros del par de cámaras propuesto.

Abstract

This work introduces the approach and implementation of an image processing system, developed to detect partially occluded people, for the purpose of assisting the search for victims in post-disaster environments as in partially collapsed buildings after a tremor.

The proposed system acquires images from the video transmission of two cameras: one sensitive to the interval of the visible electromagnetic spectrum by the human eye denoted as VIS and the second to the long-wavelength infrared band LWIR, related to thermal radiation. Both devices were installed on a base to achieve a shared field of view and establish a dot matching rule between the two pixel-level images, using projection geometry.

The camera layout described was designed to expedite and resolve the detection of people despite the conditions of the application, particularly taking into account the *diffuse distinction* of the environment of the objectives, searching for and determining analysis regions using traces of heat and the matching function between images.

It also examined the contribution that thermal imaging can make to a classifier, implemented five models based on two neural network architectures, with which you experienced readjustment and complete workouts following the methodology of local observations formulated in *Regional convolutional neural network*, to face the partial visibility of objectives in conjunction with the use of images features descriptors. This learning process was developed with a collection of VIS images and LWIR-VIS pairs of simulated people and victims.

To develop and test the pixel correlation rule between the two cameras, a perceptible calibration pattern was developed for both bands of the spectrum, and it was possible to establish the correspondence relationship with an error less than one pixel and a partial stereoscopic rectification of 75 % of the resolution.

While evaluating the classifiers, he exposed the low texture contribution of LWIR images by examining the fonts separately, but when using a space less dependent on local descriptions such as border maps, thermal projections proved to favour the classifiers and the detection stage, indicating the feasibility of developing a model that analyses the rectified four-channel image, consisting of the frames of the proposed camera pair.

Agradecimientos

Investigación de tesis realizada gracias al apoyo del programa UNAM-DGAPA-PAPIIT IT102518 "Robots móviles para la exploración e inspección de zonas con movilidad restringida", que brindó soporte en equipo electrónico y de cómputo para desarrollar este trabajo. También al Consejo Nacional de Ciencia y Tecnología por su apoyo económico que trascendente de gran ayuda para cursar los estudios de maestría.

Les agradezco de corazón a mi tutor M.I. Larry Escobar y al M.I. Yukihiro Minami por su tiempo, paciencia, consejos, instrucción confianza, orientación y oportunidades propuestas para afrontar retos de aplicación que me han brindado en todo el tiempo que he tenido la fortuna de trabajar con ellos, y que han trascendido en mi formación profesional permitiéndome adquirir habilidades y capacidades que me sirvieron para la realización del proyecto que se presenta en este documento.

Al Dr. Alfonso Gastélum por sus observaciones, recomendaciones, enseñanzas y asesoría proporcionada en el área de visión computacional que me resultaron de gran utilidad. De igual manera les agradezco a mi sinodales por su tiempo invertido en el proceso de revisión.

A mis padres y hermanas que me continuaron apoyando durante estos años que duró la maestría, aprovecho también para reconocer su esfuerzo, cariño, educación y respaldo que me han brindado y han son clave en mi vida.

A Lily que acompañó en los momentos difíciles alentándome a salir adelante y recordar los límites de lo que uno puede hacer, están en nuestra mente.

También les agradezco a mis compañeros del laboratorio de procesamiento digital de señales: Michel, Luis, Iván, Yolo y Mickey por su amistad, motivación, recomendaciones y los buenos gustos que me contagiaron por la lectura, comida y café. A mis colegas de generación Héctor y Alejandro les doy gracias por su tiempo, tolerancia, afecto y apoyo al trabajar en equipo y las múltiples conversaciones sobre nuestra área de estudio.

Y finalmente les doy gracias a todos mis camaradas del Taller de Robótica Abierta de la Facultad de Ingeniería, por su asesoría y participación en la construcción del montaje de las cámaras y por modelar para contar con imágenes de víctimas simuladas.

Prólogo

Con la finalidad de agilizar la lectura de esta tesis, en este apartado se describen una serie de observaciones y aclaraciones para favorecer el seguimiento del contenido.

En diferentes partes de este escrito se presentan acrónimos cuyo significado se indica la primera vez que aparecen en el texto, posteriormente se escriben solo las siglas pero pueden consultarse en la lista en la parte final del documento, a la que se puede llegar dando clic sobre la siglas o acrónimos según sea el caso.

También se cuenta con un *glosario* de palabras técnicas y conceptos utilizados en el desarrollo del proyecto, que son frecuentemente encontrados en la literatura de las diversas áreas de estudio involucradas. Al aparecer estos términos las primeras veces en el texto, se puede dar clic sobre la palabra y consultar su significado.

En ciertas ocasiones en el documento se indicaron citas consecutivas, que fueron expresadas en un formato de intervalo de números, por ejemplo, en [15-19] hace referencia a todos los documentos fuente enumerados en esa separación.

Y por último, en la versión digital de la tesis los números de ecuaciones, algoritmos, siglas, figuras, tablas, secciones, capítulos y referencias que se encuentran en los párrafos, cuentan con enlaces que trasladan al lugar del documento donde se encuentran al darles clic.

Contenido

Capítulo 1

Introducción

Las investigaciones de visión por computadora enfocadas en la detección de personas han presentado un auge en su desarrollo impulsado por las innovaciones en hardware y software, a causa del interés en diversas aplicaciones, como en los sistemas para detectar peatones en automóviles autónomos.

Este tipo de aplicaciones afronta factores comunes que comprometen su funcionalidad, resaltando de dichas causas los ambientes dinámicos, las múltiples posturas de un cuerpo humano, visibilidad parcial, cambios de iluminación y entornos altamente desordenados, es decir, imágenes de personas en ambientes con alto contenido de información que puede complicar su análisis.

Como respuesta a dichas circunstancias, los algoritmos para detección de formas humanas han experimentado diferentes estrategias para reducir el impacto de los factores mencionados, pretendiendo obtener buenos resultados en ambientes poco controlados, esto es que funcionen con restricciones mínimas.

En vista del favorable avance de los *clasificadores de objetos*, se ha explorado utilizarlos para buscar víctimas en situaciones de emergencia, como en avalanchas, incendios, edificaciones derrumbadas, entre otros, con el fin de agilizar diversas actividades de búsqueda y rescate urbano, referidas como USAR por sus siglas en inglés. Dicho enfoque incentivó la elaboración de este trabajo, que se decidió dedicar para detectar personas atrapadas entre escombros.

La dirección de esta investigación se decidió considerando los sistemas de instrumentación empleados por equipos de rescate dedicados a la tarea de interés, examinando en [1] su funcionamiento, ventajas, limitaciones y proyecciones de alcance. También se tomó en cuenta la factibilidad de instalar cámaras en vehículos no tripulados terrestres o aéreos, capaces de desplazarse en zonas peligrosas o de difícil acceso para una persona, con el objetivo de incrementar el alcance de búsqueda y reducir el tiempo involucrado.

El contenido de este capítulo, presenta el contexto del trabajo desarrollado para formular un detector de víctimas en entornos post-desastre, comenzando por una reseña de la documentación consultada sobre trabajos precedentes, que fundamentan la aplicación propuesta en conjunto con los objetivos que se pretenden lograr, al aplicar la metodología propuesta en la Sección 1.4.

1.1 Antecedentes

La tecnología auxiliar para encontrar víctimas atrapadas en una zona post-desastre, en su mayoría, analiza la propagación de ondas acústicas y/o electromagnéticas entre escombros desde la superficie. Un ejemplo de estos sistemas son los equipos para *detectar sonidos* [1], que tratan de rastrear señales de ayuda.

La propuesta más ambiciosa para explorar desde la superficie se ha enfocado a desarrollar *radares de signos vitales*, para detectar el movimiento del pecho durante la respiración y/o el pulso cardíaco [2]. Sin embargo, la interferencia de otras señales electromagnéticas y diferentes tipos de materiales complican la identificación y caracterización de los objetivos. No obstante, diferentes desarrolladores concluyen que en un futuro cercano estos radares tendrán una mayor eficiencia.

Debido al avance en equipos de fotografía y vídeo en conjunto con procesadores de información, se amplió el uso de cámaras digitales en actividades USAR. La portabilidad actual de dichos equipos y sistemas similares permiten utilizarlos para inspeccionar espacios confinados, con base en experiencia o por algoritmos de *aprendizaje automático*.

A causa de la dificultad inherente del campo de aplicación, también se han planteado sistemas formados por más de un tipo de señal para disponer de mayores indicios para detectar personas, como la presencia característica de dióxido de carbono o la temperatura corporal que puede examinarse con una cámara infrarroja.

Un ejemplo de estos sistemas es el propuesto por Kleiner en [3], el cual utiliza dos cámaras sensibles a diferentes intervalos del espectro electromagnético. El planteamiento consiste de un algoritmo genético basado en *Campos aleatorios de Markov* abreviado en inglés como MRF, que utiliza como variables de entrada:

- Perfiles de color en el espacio YCbCr, correspondientes a los tonos de piel de personas capturadas en imágenes con una cámara digital
- Un intervalo de la escala de grises que representa la temperatura corporal percibida por una cámara infrarroja
- Sustracciones de fondo entre imágenes sucesivas, usándolos como indicadores de movimiento
- Un mapa de gradientes o bordes por cada imagen de ambas cámaras.

Este modelo de grafo no direccionado superó en precisión y empleó menos tiempo de procesamiento que el clasificador considerado como referencia en ese momento [4], constituido por una *Máquina de soporte vectorial* [5], abreviada SVM por sus siglas en inglés, entrenada con características obtenidas de filtros en cascada definidos por la *transformada Haar* [6].

En 2010 Andriluka [7] evaluó el desempeño de cuatro algoritmos que consideró aptos para buscar *víctimas superficiales* con oclusiones parciales, utilizando una cámara monocular instalada en un Vehículo Aéreo no Tripulado denotado por sus siglas UAV en inglés. Simulando una situación de emergencia, dentro de un ambiente de oficina, adquirió imágenes de personas con diferentes obstrucciones y posturas acostadas en el suelo para realizar el análisis.

La primera etapa de evaluación consistió en comparar el desempeño de los clasificadores seleccionados usando el banco de imágenes adquirido. Al aplicar las implementaciones fuente de cada algoritmo entrenados para detectar personas de cuerpo completo o por la parte media superior, concluyó que los métodos *Picture structures*, PS [8] y *Deformable parts model*, DPM [9, 10], eran los mejores candidatos para lograr el objetivo.

Posteriormente para respaldar su afirmación, entrenó ambos clasificadores relacionando cada ***bounding box*** con el tamaño estadístico de las víctimas extraído del banco de imágenes, de manera que al rectificar las proyecciones y conocer la distancia entre la cámara y los objetivos, descartaba detecciones que no correspondían a las dimensiones de una persona.

Al realizar su segunda etapa de pruebas, observó que los errores de PS y DPM se complementaban mutuamente, por ello combinó ambos clasificadores usando un *modelo gaussiano*, logrando reducir el error de detección. En consecuencia concluyó que los modelos formados por partes descritos en [11] y retomados por Felzenswalb [12], son la mejor opción para detectar objetos con articulaciones dinámicas y ocluidos parcialmente, como las personas en ciertas condiciones.

Los descriptores formados por otros de menor magnitud como el caso de DPM, se convirtieron en tendencia de implementación en diversos detectores y clasificadores de objetos, hecho que fue notable por los trabajos que participaron en el *Reto de reconocimiento visual a gran escala* [13] ILSVRC por sus siglas en inglés, en el periodo de 2010 al 2012.

Sin embargo el proceso minucioso de entrenamiento de los métodos formados por partes no los convirtió en el esquema de referencia, sobre todo en tareas que consideraban ambientes complejos y dinámicos, como en la detección de víctimas o de peatones en calles transitadas, por lo que como alternativa se plantearon sistemas integrados por más de un tipo de clasificador.

Por ejemplo Bhuman Soni [14], en 2012 propuso un detector cuyo resultado se deriva de ponderar la respuesta de tres diferentes clasificadores: *AdaBoost* [15], *k-enésimos vecinos más cercanos* [16, 17], k-NN por sus siglas en inglés y SVM. Como datos de entrada a cada algoritmo, utilizó los descriptores *Histogramas de gradientes orientados* [18], abreviado HOG en inglés y *Speed-Up Robust Features* [19] de siglas SURF, en forma individual y combinada.

Este planteamiento contribuyó con dos conceptos relevantes para desarrollar aplicaciones con colecciones de datos no tan numerosas (como un detector de víctimas), usando estas ideas:

- ▷ Actualizar patrones de referencia en línea
  Proceso que denominó "aprendizaje incremental", porque ajustaba el modelo descriptivo que acumulará cierto factor de discordancia, comparado con el resultado de los otros clasificadores, para robustecer la decisión final del detector.
- ▷ Extender el proceso de aprendizaje
  Al incorporar a los clasificadores la capacidad de aprender al momento de su implementación, se puede reducir el tiempo de entrenamiento porque solo se consideraría un punto inicial.

A causa de los factores presentes en el ambiente de aplicación, la formación de modelos se complica, hecho al que se suma la *escasez de imágenes* por tratarse de contenido sensible, por lo que las contribuciones de Soni son destacables para el uso del proyecto.

En el mismo año Alex Krizhevsky, y sus colaboradores reavivaron el interés por las *redes neuronales* (denotadas NN en inglés), para clasificar y detectar objetos al lograr posicionar su arquitectura convolucional, conocida como *AlexNet* [20], en la lista de los primeros cinco lugares de mejor precisión en el reto ILSVRC 2012.

La formulación de *Redes neuronales convolucionales* referidas por sus siglas CNN en inglés, marcó el comienzo de metodologías de análisis alternas a los modelos de objetos basados en partes y a los sistemas de clasificadores. Un ejemplo que ocupó directamente el trabajo de Krizhevsky es el detector de víctimas presentado por Sulistijono en [21].

Esta implementación se fundamentó en la arquitectura *AlexNet*, explotando la capacidad de su bloque convolucional para definir y extraer características demostrado por [22]. Aprovechando el ***aprendizaje transferido*** (conocido como *transfer learning* en inglés) de la implementación original entrenada con la base de datos ImageNet [13], adaptó la red para detectar únicamente personas efectuando las siguientes modificaciones:

1. Retiró la última capa totalmente conectada de neuronas denominada fc_8, y asignó a su predecesora fc_7 la función de activación adecuada para proporcionar la probabilidad de detección.
2. Entrenó específicamente solo las dos capas de neuronas totalmente conectadas fc_6 y fc_7, con imágenes de personas extraídas de las bases de datos VOC 2010 [23] e IDV-50, esta última formada por los autores [21] con escenas de víctimas reales encontradas en Internet.

Para entrenar la red y detectar regiones de las imágenes para analizar con el clasificador, utilizó el algoritmo de *Búsqueda selectiva* [24] extrayendo áreas de 227×227 pixeles.

Como resultados comprobaron que *AlexNet* puede describir y detectar personas usando el entrenamiento inicial usando el dataset *ImageNet*, sin embargo reportaron problemas al probar con imágenes donde las personas estaban cubiertas por polvo y/o sin alguna parte característica del cuerpo visible, resaltando la necesidad de un estudio más especifico para escenarios post-desastre.

En los años posteriores se exploró la capacidad de las *redes neuronales* para detectar, clasificar y segmentar objetos, logrando incrementar su precisión de acuerdo a los reportes anuales de ILSVRC. Las nuevas arquitecturas desarrolladas también se aplicaron para la detección de víctimas, como en el planteamiento para buscar sobrevivientes después de una avalancha de nieve, publicado por Bejiga en [25].

Esta implementación utilizó la red *GoogLeNet* [26] entrenada con el dataset ImageNet, para extraer representaciones características de una colección de imágenes y vídeos grabados desde un UAV de víctimas simuladas y reales, y después un bloque externo a la red analizaba el mapa de características con un clasificador binario SVM. Adicionalmente para procesar transmisiones de vídeo desde el UAV, incorporó como etapa extra a su sistema de detección un *Modelo oculto de Markov* para mejorar la precisión del clasificador, para considerar las decisiones previas.

Bejiga y sus colaboradores en su etapa de experimentación analizaron diferentes resoluciones de vídeo, obteniendo índices de precisión entre el 70 y 94 % logrando el factor más alto en la máxima resolución, a cambio de un mayor tiempo de procesamiento.

La correlación entre los trabajos mencionados que han aplicado redes neuronales, es que consideran la información directa de los pixeles como datos de entrada para detectar objetos. En los dos casos presentados [21, 25], la redes se utilizaron como *extractores de características*, sin embargo, en otros estudios se han empleado redes usando diferentes datos de entrada y metodologías para modelar los objetos.

Ouyang, por ejemplo, formuló un *detector de peatones* usando una CNN dedicada a modelar los objetivos como DPM, considerando oclusiones y variaciones de posición de las extremidades del cuerpo [27]. Como datos de entrada a la red usó un arreglo de tres matrices extraídas de regiones de 84×28 pixeles, en donde cada matriz consistió de:

1. El canal de luminiscencia Y al convertir la región por analizar del espacio de colores RGB al YCbCr
2. Un mapa de intensidades de pixeles integrado por cuatro secciones, que corresponden a cada canal del espacio YCbCr de la región de interés escalada y ajustada a 42×14 pixeles. El sector restante es completada con una matriz de ceros
3. Un mapa de bordes integrado por cuatro secciones de 42×14 pixeles, donde tres son los bordes calculados de cada canal Y, Cb y Cr, y el cuarto mapa se forma de los máximos registrados elemento por elemento, de cada uno de los mapas anteriores.

Al evaluar el algoritmo después de entrenarlo con un conjunto de imágenes de peatones parcial y totalmente visibles, Ouyang reportó un desempeño superior respecto a otros detectores que usaron las mismas bases de datos. Este planteamiento fue uno de los primeros en utilizar una red neuronal para analizar objetos con obstrucciones, con una idea similar a DPM o PS usando arreglos de entrada a la red formados por contrastes y mapas de bordes.

Uno de los objetivos de ILSVRC era formular un algoritmo que superara el 5.1 % del *error de rendimiento humano* en la tarea de clasificación y detección de objetos en imágenes [28].

AlexNet y GoogLeNet fueron arquitecturas reconocidas por los resultados en este reto, porque lograron identificar objetivos con oclusiones impulsando el desarrollo de mejores planteamientos que lograron superar el 5.1 % de error, como la *Red neuronal convolucional regional* [29] de siglas R-CNN, que se explicará en el Capítulo 4.

1.2 Planteamiento del problema

La finalidad de este proyecto es analizar el rendimiento y viabilidad de implementar un algoritmo de visión por computadora, para detectar víctimas no superficiales y parcialmente visibles en entornos post-desastre. Este tipo de sistemas auxiliares se han instalado en vehículos de exploración terrestre y/o aéreos, así como en herramientas o artefactos que permitan acercar el equipo a la zona de emergencia por el personal de rescate.

La mayoría de los trabajos antecedentes que se presentaron en la Sección 1.1, son aplicaciones desarrolladas en su mayoría para exploración terrestre, considerando como fundamento los algoritmos presentados en ILSVRC y detectores de peatones en ambientes dinámicos y de alto desorden [27, 30, 31], por la mediana similitud de las condiciones de trabajo.

Para lograr el propósito de detectar víctimas en entornos reales aplicando un algoritmo de visión computacional, una de las consideraciones más trascendentes para desarrollar un clasificador de imágenes es:

> *El aleatorio punto de observación y la información que se pueda captar por el equipo de visión, al buscar en una zona de emergencia como en una construcción derrumbada.*

A este punto de atención, también se suman otros factores inherentes que deben tomarse en cuenta para formular una metodología capaz de funcionar en escenarios reales. Algunos de estos agentes reportados en [32] son los siguientes:

- Escasa o nula iluminación
- Modelar posturas del cuerpo poco comunes para los detectores de personas
- Visibilidad comprometida de las víctimas por obstrucciones de otros objetos
- Presencia de polvo en el ambiente y sobre las personas, tanto en partes descubiertas por la ropa como en esta
- Observar partes del cuerpo heridas que modifiquen su percepción por el clasificador.

En la Figura 1.1 se muestran algunas imágenes que ejemplifican los factores y situaciones mencionados que se tomarán en cuenta para hacer un planteamiento del sistema.

Diferentes aplicaciones con enfoques semejantes a la *detección de víctimas*, han planteado enfrentar algunos de los factores citados utilizando cámaras perceptibles a la *radiación infrarroja*, abreviada IR, con interés particular en equipos con sensores sensibles al intervalo de *longitud de*

Figura 1.1. Imágenes ejemplo de víctimas en una situación real [21].

onda larga abreviado LWIR en inglés, porque es proporcional a la temperatura de los cuerpos.

Por ejemplo, en [3, 33–36] se formularon metodologías complementarias de clasificadores, utilizando cámaras IR y VIS, fundamentadas en algoritmos que procesan imágenes VIS. En estos proyectos los equipos de adquisición de vídeo comparten un *campo de visión*, para observar casi la misma escena y proceder a procesar o analizar los objetivos usando la información que aporta cada sensor.

Con base en los resultados de los trabajos citados, se puede considerar factible la propuesta de aplicar un *sistema de cámaras de sensibilidad combinada* para la detección de víctimas en entornos post-desastre, a reserva del costo. Al contar con estos equipos, *otras aplicaciones relacionadas* con actividades USAR también *pueden ser puestas en operación* en el mismo sistema, como el trabajo de Jiménez en [37] que estima el pulso cardíaco a distancia.

Al contar con mayor información del objeto de interés a detectar, las metodologías de análisis expuestas podrían aumentar su porcentaje de detección y superar resultados confusos, por lo que considerando la documentación presentada en las Secciones 1.1 y 1.2 se plantearon los siguientes objetivos, para desarrollar este trabajo.

1.3 Objetivos del proyecto

Objetivo general

Plantear e implementar un algoritmo detector de personas orientado a la detección de víctimas en situaciones de emergencia, analizando imágenes de dos cámaras diferentes y considerando determinadas circunstancias probables en entornos reales.

El sistema propuesto contará con las herramientas necesarias para poder llegar a trabajar con una transmisión de vídeo de ambas fuentes, tomando en cuenta que pueda instalarse en un vehículo de exploración terrestre.

Objetivos específicos

- Relacionar los campos de visión de una cámara infrarroja y una de visión normal, para analizar ambas fuentes de información a partir de una correspondencia de coordenadas.
- Analizar algoritmos de clasificación de personas que puedan detectar víctimas en entornos post desastre, manejando oclusiones y múltiples posturas.
- Implementar un clasificador de personas que tenga la capacidad de funcionar en tiempo real.
- Hacer un banco de imágenes con el sistema de cámaras propuesto, de personas parcialmente ocluidas que se aproximen a una situación real.

1.4 Metodología

Para cumplir con los propósitos y metas descritas en la Sección 1.3, se planteó el siguiente procedimiento de acuerdo al orden de requerimientos de la aplicación.

I Adquisición de imágenes

Se decidió trabajar con dos cámaras de diferente sensibilidad; una al intervalo visible por los humanos del espectro electromagnético VIS y otra a la banda infrarroja de longitud de onda larga LWIR por los siguientes argumentos:

1. Se puede obtener una proyección de la escena con escasa iluminación
2. Es posible agilizar la búsqueda de objetivos en una escena, por causa de la irradiación de calor de los cuerpos que resalta su silueta del entorno
3. Las texturas de las imágenes IR pueden complementar las que normalmente se perciben con cámaras VIS, por lo que pueden superarse malas detecciones por heridas o falta de contraste de la ropa con el fondo
4. Después de detectar a una víctima y si las circunstancias de observación lo permiten, con la cámara LWIR es posible estimar el pulso cardíaco y/o la frecuencia de respiración, implementando una inspección visual o algún algoritmo como [37].

A estos motivos se agrega la línea de trabajo antecedente de [1, 37, 38] vinculada a este proyecto y para experimentar las mejoras de resultados que puede lograr analizar proyecciones de diferentes tipos de sensores.

Para lograr el *campo de visión* compartido entre ambas cámaras monoculares, se colocarán en un soporte con una configuración semejante a un sistema estereoscópico, procurando a nivel macroscópico, alinear horizontalmente las lentes.

Después con fundamento en la teoría de proyección de perspectiva, se determinará la relación de correspondencia entre pixeles de ambas imágenes, prescindiendo del uso de puntos característicos para aplicarse en línea.

II Estudio de los descriptores de formas antropomórficas

En virtud de la información de las dos cámaras consideradas, se examinarán los métodos para describir proyecciones de personas, reportados como funcionales en diversos trabajos antecedentes que han considerado factores semejantes a los mencionados en la Sección 1.2. Con esto se pretende analizar los espacios de representación de características, y observar cuál es más conveniente utilizar tomando en cuenta que las imágenes son de diferente tipos de texturas.

III Selección de un clasificador

De acuerdo a las metodologías consultadas en proyectos afines, los factores inherentes a la aplicación mencionados en la Seccion 1.2 y los objetivos planteados se elegirá uno de los algoritmos de clasificación de imágenes diseñado para afrontar obstrucciones de los objetivos de interés, como fundamento de implementación del sistema de detección propuesto integrado por cuatro etapas:

1. Búsqueda de regiones de imagen candidatas para analizarse.
2. Acondicionamiento y estructuración de señales para ingresar al clasificador
3. Clasificación de la señal de entrada de acuerdo al algoritmo seleccionado.
4. Estimación del porcentaje de similitud y presentación de resultados de detección.

La detección de las personas en las imágenes se realizará proponiendo regiones de análisis utilizando información estadística del conjunto de muestras de entrenamiento, segmentación por *búsqueda selectiva* y/o por fuentes de calor detectadas por la cámara LWIR. En cualquiera de estos casos, la localización del objeto en alguna de las dos imágenes será posible por la correspondencia de coordenadas entre ambas cámaras.

Después de seleccionar todos los métodos a implementar en conjunto, se procederá a desarrollar todo lo necesario para realizar los experimentos del sistema propuesto y reportar sus resultados.

1.5 Estructura de la tesis

Con el objetivo de proporcionar un esquema general de este documento, a continuación se reseñan brevemente el contenido de los capítulos de esta tesis.

En el desarrollo de este primer Capítulo, se presentó el tema de interés del proyecto orientado a la detección de víctimas en entornos post-desastre analizando imágenes, para el cual se reportaron brevemente diversos trabajos antecedentes que funcionaron como base para plantear al sistema de detección propuesto, en conjunto con los objetivos deseados y una metodología propuesta para lograrlos.

Respecto al orden de procedimientos de la Sección 1.4, el ***Capítulo 2*** contiene el marco teórico relacionado con la correspondencia de coordenadas entre las cámaras VIS e IR. En específico se presenta el *modelo de proyección* especifica y la metodología de calibración de una cámara

monocular, que son fundamentos para establecer una regla de correspondencia entre pixeles de las dos imágenes.

Posteriormente, en el ***Capítulo 3*** se explican los métodos más utilizados para describir la fisonomía de las personas proyectadas en una imagen. Los *descriptores* que se exponen son reportados como los más adecuados en diversos trabajos, que han considerado factores de implementación similares a los involucrados en este proyecto como los citados en la Sección 1.1.

En el ***Capítulo 4*** se revisa el fundamento de los algoritmos de *aprendizaje automático* utilizados para clasificar imágenes, con el objetivo de establecer el contexto necesario para presentar las metodologías para detectar personas que se consideraron como más apropiadas para la aplicación. Y en la parte final de este apartado se describen los indicadores más utilizados para evaluar el desempeño de los clasificadores.

La implementación del sistema propuesto para detectar víctimas en entornos post-desastre se describe en el ***Capítulo 5***. Durante el desarrollo de este capítulo, se exponen las actividades involucradas en el desarrollo del proyecto, incluyendo las especificaciones de hardware y software que se ocuparon en diferentes tareas para poder evaluar la propuesta planteada.

Las pruebas experimentales de los módulos del sistema y los resultados obtenidos están registrados y especificados en el ***Capítulo 6***. Y por último, el ***Capítulo 7*** comprende las conclusiones del trabajo, que consta del análisis de los resultados obtenidos en las diferentes pruebas realizadas y las recomendaciones del trabajo que se podría desarrollar para mejorar los resultados.

Capítulo 2

Proyección, calibración y relación de pixeles entre dos cámaras monoculares

Las imágenes son representaciones de los objetos dentro de un *campo de visión*, determinado por una lente o conjunto de ellas donde convergen haces de luz, provenientes de diferentes fuentes, hacia una pantalla, receptor o sensor perceptible a cierto rango del espectro electromagnético.

Todos los equipos de adquisición de imágenes y vídeo que conocemos actualmente, son resultado del estudio de la Óptica y por el desarrollo tecnológico de hardware y software. Modelos y conceptos específicos de esta rama de la Física, son involucrados en aplicaciones de visión artificial y procesamiento de imágenes, porque modifican o distorsionan la información por analizar.

En virtud del planteamiento y metodología expuestos en las Secciones 1.2 y 1.4, el contenido de este Capítulo está dedicado al marco teórico relacionado con la caracterización de las cámaras VIS e IR, para corregir efectos de distorsión y determinar la correspondencia geométrica entre las imágenes adquiridas por cada equipo.

La estructura de este Capítulo es la siguiente; en la Sección 2.1 se describe el *modelo de proyección* de imágenes de la cámara *pinhole*, definido principalmente por los parámetros característicos del equipo. Estos datos son determinados de forma experimental junto con los factores de *distorsión óptica*, aplicando el algoritmo presentado en la Sección 2.2 a cada una de las cámaras. Los conceptos presentados son el fundamento de la Sección 2.3 que expone dos metodologías para establecer la correspondencia entre coordenadas de las dos fuentes de imagen.

2.1 Modelo de proyección de perspectiva

2.1.1 Proyección normalizada

La proyección de puntos de un espacio tridimensional (3D) a uno bidimensional (2D), aplicado a imágenes es descrito por el modelo de ***cámara pinhole***, representado por una caja cúbica con un pequeño orificio, ubicado en el centro de una de sus caras como se muestra en la Figura 2.1.

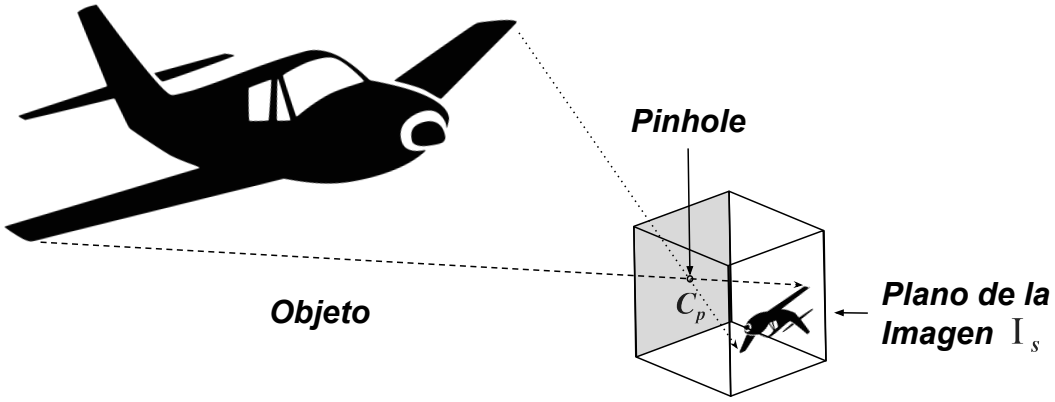


Figura 2.1. Ilustración del modelo de la cámara de pinhole.

A través del *pinhole* pasan al interior de la cámara, rayos electromagnéticos de una región del mundo exterior, procedentes de fuentes de radiación y reflexiones de estas en el medio. Esto forma una representación invertida de los diferentes cuerpos presentes al frente de la cámara, en la cara opuesta a la ubicación C_p del orificio, denominada como *plano de la imagen* I_s [39].

Para analizar el modelo sin la inversión de imagen, se intercambia de posición I_s y C_p como se ilustra en la Figura 2.2. La proyección de un punto $\dot{Q}$ de algún objeto 3D dentro del campo de visión, comienza al representar su posición en el sistema de referencia de la cámara, aplicando la descripción de *movimiento de un cuerpo rígido* [40, 41]

$$\dot{Q}' = \dot{Q} \cdot \mathbf{R}_3 + t \tag{2.1}$$

donde $\mathbf{R}_3$ cuantifica el movimiento de *rotación*, determinado por un cambio de orientación debido al giro en cada eje coordenado

$$\mathbf{R}_3 = R_x \cdot R_y \cdot R_z = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{2.2}$$

en donde cada matriz factor está definida como se muestra a continuación

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_x) & -\sin(\theta_x) \\ 0 & \sin(\theta_x) & \cos(\theta_x) \end{bmatrix} \tag{2.3a}$$

$$R_y = \begin{bmatrix} \cos(\theta_y) & 0 & \sin(\theta_y) \\ 0 & 1 & 0 \\ -\sin(\theta_x) & 0 & \cos(\theta_y) \end{bmatrix} \tag{2.3b}$$

$$R_z = \begin{bmatrix} \cos(\theta_z) & -\sin(\theta_z) & 0 \\ \sin(\theta_z) & \cos(\theta_z) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.3c}$$

El orden de los factores del producto (2.2), corresponde a la definición del sistema de ejes ortogonales, por lo que no es conmutativo. Y el cambio de posición es declarado en el vector de traslación, denotado como

$$t = [t_x,\ t_y,\ t_z]^T \tag{2.4}$$

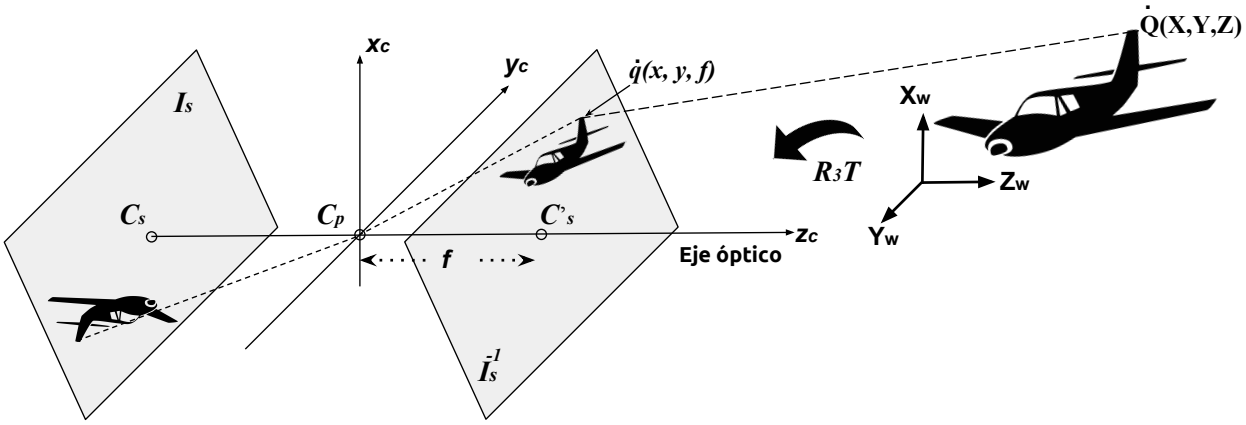


Figura 2.2. Diagrama de proyección del modelo de cámara pinhole.

Este cambio de referencia, se representa en la Figura 2.2. Después $\dot{Q}'$ es proyectado en I_s, al aplicar equivalencia de triángulos semejantes obteniendo

$$\frac{X'}{Z'} = \frac{x}{f} \quad \Rightarrow \quad x = f \cdot \frac{X'}{Z'} \tag{2.5a}$$

$$\frac{Y'}{Z'} = \frac{y}{f} \quad \Rightarrow \quad y = f \cdot \frac{Y'}{Z'} \tag{2.5b}$$

estas correspondencias, comúnmente son expresadas en notación matricial de la siguiente manera

$$\dot{q} = \begin{bmatrix} x \\ y \end{bmatrix} = \frac{f}{Z'} \cdot \begin{bmatrix} X' \\ Y' \end{bmatrix} \tag{2.5c}$$

donde f es la *distancia focal* que separa I_s y C_p observada en la Figura 2.2. El proceso de proyección descrito en (2.5), es sintetizado en una expresión lineal, al usar ***coordenadas homogéneas*** en lugar de cartesianas. La transformación se define para un punto de n dimensiones $\dot{g} = [g_0,\ \dots,\ g_{n-1}]^T$, por la función $hom\,(\cdot)$ como

$$\tilde{\dot{g}} = hom\,(\dot{g}) = [\tilde{g}_0,\ \dots,\ \tilde{g}_{n-1},\ \tilde{g}_n]^T = s \cdot [g_0,\ \dots,\ g_{n-1},\ 1]^T \tag{2.6a}$$

y la relación inversa se define

$$\dot{g} = hom^{-1}\left(\tilde{\dot{g}}\right) = [g_0,\ \dots,\ g_{n-1}]^T = \frac{s^{-1}}{\tilde{g}_n} \cdot [\tilde{g}_0,\ \dots,\ \tilde{g}_{n-1}]^T \quad |\tilde{g}_n \neq 0 \tag{2.6b}$$

en las cuales $s \epsilon \mathbb{N}$ es un factor de escala que determina la equivalencia de dos o más puntos homogéneos, con un punto cartesiano [41], es decir

$$\tilde{\dot{g}} \equiv s \cdot \tilde{\dot{g}} \tag{2.6c}$$

Con base en (2.6a), al convertir $\dot{Q}$ a coordenadas homogéneas es posible expresar (2.1) como

$$\widetilde{\dot{Q}'} = \widetilde{R_3 T} \cdot \widetilde{\dot{Q}} \quad \Rightarrow \quad \begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.7a}$$

donde

$$\widetilde{R_3 T} = M_t \cdot hom\,(\mathbf{R}_3) = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.7b}$$

en la cual M_t es la conversión de (2.4) a coordenadas homogéneas, transformando el vector a una matriz de acuerdo a la subsecuente equivalencia

$$\widetilde{\dot{Q}_t} = hom\,(\dot{Q} + t) = \begin{bmatrix} X + t_x \\ Y + t_y \\ Z + t_z \\ 1 \end{bmatrix} \equiv M_t \cdot hom\,(\dot{Q}) = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.7c}$$

de manera que al expresar la proyección $\dot{q}$ al dominio homogéneo, considerando (2.7a), la correspondencia (2.5c) y la equivalencia (2.6c) para $s = Z'$ se obtiene que

$$\tilde{q} = hom\,(\dot{q}) = \begin{bmatrix} fX'/Z' \\ fY'/Z' \\ 1 \end{bmatrix} \equiv \begin{bmatrix} fX' \\ fY' \\ Z' \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = M_p \cdot \widetilde{\dot{Q}'} \tag{2.8}$$

en donde M_p es la ***matriz de proyección*** que relaciona la dimensión 3D con 2D en el dominio homogéneo. Al sustituir (2.7a) en (2.8) y considerar $f = 1$, se obtiene el modelo ideal de ***proyección normalizada*** [41] de la cámara *pinhole*

$$\tilde{q} = \mathbf{R_3 T} \cdot \widetilde{\dot{Q}} \tag{2.9a}$$

expresado en coordenadas cartesianas, aplicando (2.6b) como

$$\dot{q} = hom^{-1}\left(\mathbf{R_3 T} \cdot \widetilde{\dot{Q}}\right) = [x,\, y]^T \tag{2.9b}$$

en donde $\mathbf{R_3 T}$ define la posición del punto $\dot{Q}$ respecto a la localización de la cámara, proyectándose en el plano de la imagen de acuerdo a la *matriz de proyección*, de la siguiente manera

$$\mathbf{R_3 T} = M_p\big|_{f=1} \cdot \widetilde{R_3 T} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \tag{2.10}$$

los elementos de la matriz (2.10) son denominados ***parámetros extrínsecos*** y estos son únicos para cada proyección de la cámara, porque representan el cambio de perspectiva de cada imagen.

2.1.2 Distorsiones ópticas

El modelo (2.9) está definido únicamente por *geometría de proyección*. Pero las cámaras reales, utilizan por lo menos una lente para converger la radiación a la película o sensor fotosensible (*plano de la imagen*), posicionando el monóculo en el lugar del *pinhole*, lo cual modifica la distancia entre I_s y C_p debido al punto focal de la lente como se muestra en la Figura 2.3.

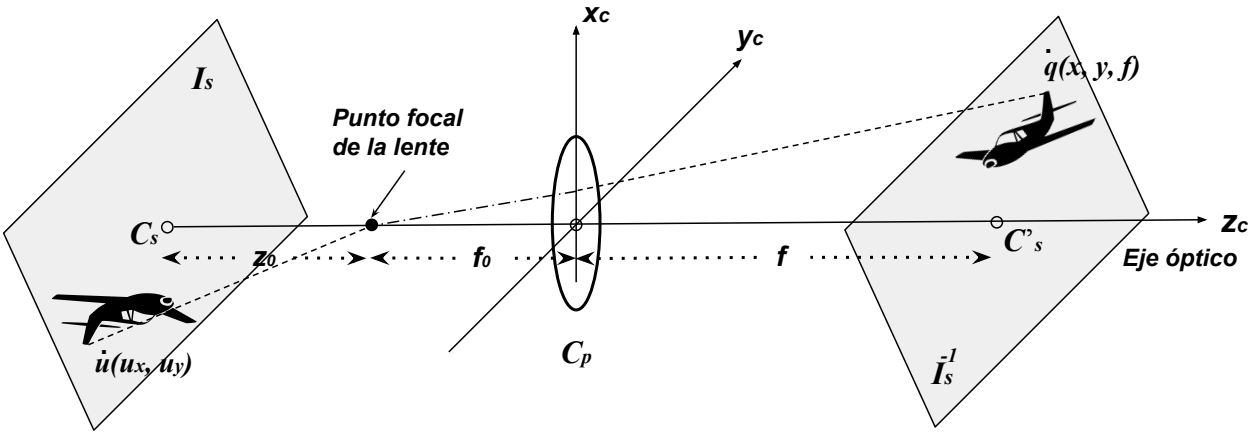


Figura 2.3. Diagrama de proyección monocular.

La curvatura y fabricación de las lentes, introducen deformaciones ópticas en la proyección de cada punto definido por (2.9). Las dos distorsiones más perceptibles en la imagen, son modeladas y compensadas como se describe a continuación.

Distorsión radial

Es el desplazamiento de coordenadas en la proyección, alejándose o acercándose al centro del *plano de la imagen*, debido a la geometría curva de la lente. Al distanciarse esta alteración se denomina *distorsión Barrel* y el caso opuesto se llama *distorsión pincushion* [42].

La presencia de esta deformación provoca el redondeo de bordes y la curvatura de rectas, que en realidad no lo están. Estos efectos son más perceptibles en los límites de la imagen, como puede observarse en la simulación de la Figura 2.4, donde se muestra el impacto de los dos tipos descritos anteriormente.

La ***distorsión radial*** está en función de la distancia Euclidiana r_j que existe entre cualquier punto $\dot{q}_j$ y el centro del *plano de la imagen* C_s, en principio alineado sobre el eje óptico junto con el centro de la lente. Este efecto de distorsión es cuantificado en [43] como

$$\mathfrak{D}_r\left(r_j, k\right) = \sum_{i=1}^{l} k_i \cdot {r_j}^{2i} \qquad i \epsilon \mathbb{N}; i = 1, 2 \ldots, l \tag{2.11}$$

en donde l es la cantidad de *coeficientes de distorsión radial* k_i a utilizar en el modelo, demostrándose experimentalmente por Fryer que dos coeficientes son suficientes, o tres en caso

(a) Efecto de la distorsión Barrel. (b) Efecto de la distorsión Pincushion.

Figura 2.4. Simulación de los efectos de distorsión radial.

de trabajar con lentes de gran angular [44]. Los valores de cada k_i son calculados en el proceso de calibración, descrito en la Sección 2.2.

Distorsión por descentralización

Idealmente, el centro de curvatura superficial de la lente debe ser colineal con el centro del *plano normalizado*. Sin embargo, en los procesos de ensamble de las cámaras es posible que exista una diferencia entre sus posiciones, distorsionando la proyección.

A este efecto se le conoce como ***distorsión por descentralización o tangencial***, descrito inicialmente por Conrady en [45] y concluido por Brown en [46]. La ponderación usada en la actualidad de esta deformación, fue publicada por Fryer en [47] y consiste en lo siguiente

$$\mathfrak{D}_c\left(r_j, p\right) = \begin{bmatrix} \mathfrak{D}_{c_x} \\ \mathfrak{D}_{c_y} \end{bmatrix} = \begin{bmatrix} 2p_1xy + p_2\left({r_j}^2 + 2x^2\right) \\ p_1\left({r_j}^2 + 2y^2\right) + 2p_2xy \end{bmatrix} \tag{2.12}$$

donde p_1 y p_2 son los *coeficientes de distorsión tangencial*, r_j es la distancia radial entre C_s y las coordenadas corresponden a la proyección normalizada de un punto $q_j = [x, y]^T$.

Corrección de distorsión óptica

Al cuantificar la dos deformaciones mas significativas atribuidas a la lente (2.11) y (2.12), se pueden rectificar dichos efectos directamente sobre las coordenadas de cualquier punto j definido por el modelo (2.9b) como

$$q_{j_{sd}} = q_j \cdot \left[1 + \mathfrak{D}_r\left(r_j, k\right)\right] + \mathfrak{D}_c\left(r_j, p\right) \tag{2.13a}$$

o escrito en forma desarrollada considerando tres coeficientes de *distorsión radial* $l = 3$

$$\begin{bmatrix} x_{j_{sd}} \\ y_{j_{sd}} \end{bmatrix} = \begin{bmatrix} x_j \\ y_j \end{bmatrix} \cdot \begin{bmatrix} 1 + k_1r_j^2 + k_2r_j^4 + k_3r_j^6 \\ 1 + k_1r_j^2 + k_2r_j^4 + k_3r_j^6 \end{bmatrix} + \begin{bmatrix} 2p_1x_jy_j + p_2\left(r_j^2 + 2x_j^2\right) \\ p_1\left(r_j^2 + 2y_j^2\right) + 2p_2x_jy_j \end{bmatrix} \tag{2.13b}$$

los coeficientes de ambas distorsiones, por fines prácticos suelen expresarse en un solo vector integrado de la siguiente forma

$$\mathfrak{d}_{rc} = [k_1, k_2, k_3, p_1, p_2] \tag{2.14}$$

representa los *coeficientes de distorsión óptica* para rectificar una imagen [48, 49].

2.1.3 Proyección específica

Para formar la imagen de los objetos al frente de la cámara, los puntos proyectados por (2.9) en el *plano de la imagen* I_s, son escalados y sesgados por uno de los factores que se obtienen al descomponer la *matriz de proyección*

$$M_p = M_{p_f} \cdot M_{p_0} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{2.15}$$

Retirando la condición $f = 1$, la correspondencia de puntos 3D con 2D en coordenadas homogéneas se determina por M_{p_0} y los parámetros que escalan y sesgan son incluidos en M_{p_f} como

$$\mathrm{K} = \begin{bmatrix} s_x & s_\theta & c_x \\ 0 & s_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} fs_x & fs_\theta & c_x \\ 0 & fs_y & c_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \alpha & \gamma & c_x \\ 0 & \beta & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2.16}$$

en donde s_x y s_y son las escalas de cada eje del sistema de referencia, s_θ es el factor de oblicuidad ocasionado por la posible distorsión diagonal de la película o sensor fotosensible, y c_x, c_y son las coordenadas del centro de I_s, denotado como C_s. K es denominada como ***matriz intrínseca*** y sus elementos son los ***parámetros intrínsecos*** característicos para cada cámara.

Con base en la descomposición (2.15) y considerando $f \neq 1$ en la matriz (2.16), se retira la normalización de la expresión (2.9a), definiendo consecuentemente el ***modelo de proyección de perspectiva*** en *coordenadas homogéneas* para un punto $\dot{Q}_j$

$$\widetilde{ui_j} = \mathrm{K} \cdot \mathbf{R}_3\mathbf{T} \cdot \widetilde{Q_j} \tag{2.17a}$$

aplicando (2.6b) también es expresado en coordenadas cartesianas como

$$ui_j = hom^{-1}\left(\mathrm{K} \cdot \mathbf{R}_3\mathbf{T} \cdot \widetilde{Q_j}\right) = \left[u_x, u_y\right]^T \tag{2.17b}$$

considerando distorsión óptica nula. Al involucrar dichas alteraciones, rectificadas en (2.13a) la proyección de puntos en una imagen, se obtiene de la siguiente manera

$$ui_j = \mathrm{K}' \cdot hom\left(q_{j_{sd}}\right) = \begin{bmatrix} \alpha & \gamma & c_x \\ 0 & \beta & c_y \end{bmatrix} \cdot \begin{bmatrix} x_{j_{sd}} \\ y_{j_{sd}} \\ 1 \end{bmatrix} = \begin{bmatrix} u_x \\ u_y \end{bmatrix} \tag{2.17c}$$

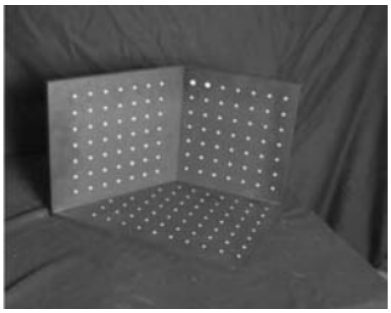
Las transformaciones (2.17), convierten las coordenadas de cada punto proyectado a *pixeles*, con y sin distorsión respectivamente [41]. Todos los parámetros involucrados en el *modelo de proyección de perspectiva*, son calculados por algoritmos de calibración que se describen en la Sección 2.2, considerando el desarrollo presentado.

2.2 Calibración de una cámara monocular

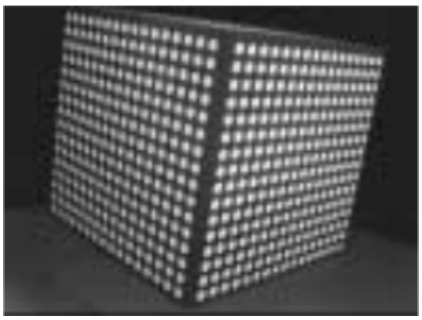
Es el procedimiento empleado para definir principalmente los parámetros *intrínsecos*, y de ellos estimar los *extrínsecos* del modelo de *proyección de perspectiva*, usando referencias del espacio 3D, incluyendo en algunos algoritmos los coeficientes de distorsión óptica.

Las diferentes metodologías que existen para calibrar una cámara monocular, se distinguen por utilizar o prescindir de un objeto o conjunto de insignias específicas, como guías de proyección. Sin embargo, a pesar de elegir un proceso de acuerdo a la aplicación, los algoritmos con mayor desarrollo son aquellos que usan un marco de referencia finita, denominado como ***patrón de calibración*** [42, 50].

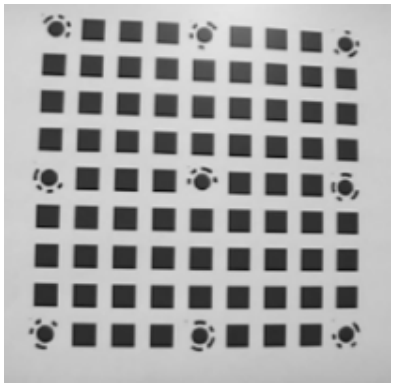
En el planteamiento resaltado, la correspondencia 3D → 2D se obtiene a partir de N puntos de referencia $\dot{Q}_j$ y sus respectivas proyecciones ${u_j}^k$, de una o k-imágenes del *patrón de calibración*, de acuerdo al método seleccionado. En la Figura 2.5 se muestran cuatro diferentes ejemplos de estas guías.



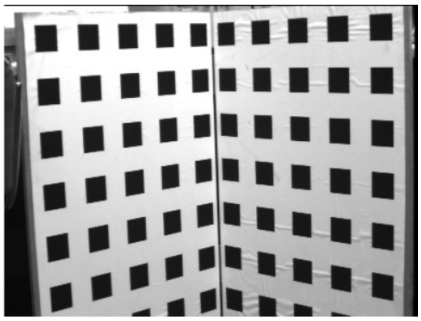
(a) Sistema de referencia formado por planos ortogonales, Quan [51].



(b) Cubo de calibración, con mallas de puntos, Heikkilä [49].



(c) Patrón plano de referencia, Zhang [50]



(d) Patrón de calibración denominado malla de Tsai [39, 52]

Figura 2.5. Ejemplos de patrones de calibración utilizados en diferentes trabajos y proyectos de investigación

Los sistemas de referencia usados para estimar los parámetros del *modelo de cámara finita* (2.17), usualmente están formados por poliedros, un plano o arreglos de dos o tres planos con orientaciones diferentes. En las caras o superficies de estos objetos, se graban o marcan los puntos de referencia $\dot{Q}_j$ que suelen ser centros de figuras geométricas, intersecciones de líneas paralelas y perpendiculares, conjuntos de rectas o señales singulares como códigos.

El propósito de la aplicación y tipo de algoritmo a utilizar para calibrar, determinan la precisión necesaria para manufacturar el *patrón de calibración*. Esta relación dio lugar a planteamientos que prescinden de un objeto especifico, denominados como procesos de *auto calibración* y también a métodos tolerantes a mediciones precisas en la fabricación del sistema de referencia [50].

Al usar un patrón, las proyecciones ${u_j}^k$ de los puntos de referencia $\dot{Q}_j$, se obtienen aplicando métodos de detección definidos por la guía 3D, dando paso a la etapa común de los algoritmos de calibración de este enfoque, que se describe a continuación.

2.2.1 Estimación de la matriz de homografía

El modelo de *cámara finita* (2.17) tiene como núcleo de transformación la *matriz de proyección*, caracterizada por los *parámetros intrínsecos* y *extrínsecos* con base en la factorización (2.15). La correspondencia (2.17a) en *coordenadas homogéneas* es simplificada por la ***matriz de homografía*** de manera que puede reescribirse como

$$\mathbf{H}_k = \mathrm{K} \cdot \mathbf{R}_3\mathbf{T}_k = \begin{bmatrix} \alpha & \gamma & c_x \\ 0 & \beta & c_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix}_k = \begin{bmatrix} H_{11} & H_{12} & H_{13} & H_{14} \\ H_{21} & H_{22} & H_{23} & H_{24} \\ H_{31} & H_{32} & H_{33} & H_{34} \end{bmatrix}_k \tag{2.18}$$

en donde el subíndice k indica la imagen asociada a la matriz. Al reunir las variables necesarias del proceso de proyección, el modelo (2.17a) puede reescribirse en términos de (2.18) de la siguiente manera

$$\begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix}_j = \begin{bmatrix} H_{11} & H_{12} & H_{13} & H_{14} \\ H_{21} & H_{22} & H_{23} & H_{24} \\ H_{31} & H_{32} & H_{33} & H_{34} \end{bmatrix}_k \cdot \begin{bmatrix} X_j \\ Y_j \\ Z_j \\ 1 \end{bmatrix} \tag{2.19}$$

Calcular $\mathbf{H}_k$ comúnmente es la etapa inicial de la mayoría de procedimientos de calibración, que utilizan una guía de referencia. A pesar de que algunos métodos hacen simplificaciones de la *matriz de homografía* o la definen de acuerdo a (2.9a), la mayoría de las técnicas implementa un mismo planteamiento, sin importar las diferentes suposiciones para determinarla o el tipo de *patrón de calibración* usado [39].

Esta fase compartida se debe a la condensación de los parámetros (2.16) y (2.10) en un solo arreglo, que se define de acuerdo al método utilizado para *calibrar la cámara*. El Algoritmo 2.2.1 resume el cálculo de $\mathbf{H}_k$ considerando el modelo (2.19) para una imagen I_k.

Algoritmo 2.2.1: Estimación estándar de la matriz de homografía.

Entrada(s): $\mathrm{I}_k \Rightarrow$ Imagen del *patrón de calibración*

$\mathcal{Q} = \left[\dot{Q}_1, \dot{Q}_2, \ldots, \dot{Q}_j, \ldots, \dot{Q}_N\right] \Rightarrow N$-puntos de referencia, donde $\dot{Q}_j = \left[X_j, Y_j, Z_j\right]^T$

$\mathcal{u} = \left[\dot{u}_1, \dot{u}_2, \ldots, \dot{u}_j, \ldots, \dot{u}_N\right] \Rightarrow$ Proyecciones de los puntos de referencia, donde $\dot{u}_j = \left[u_{x_j}, u_{y_j}\right]^T$

Salida(s): Matriz de homografía $\mathbf{H}_k$ (2.18)

1 ▷ Estimación inicial
2 $N_{\mathcal{Q}}, \mathcal{Q}_\eta \leftarrow normalizar\_puntos\,(hom\,(\mathcal{Q}))$
3 $N_{\mathcal{u}}, \mathcal{u}_\eta \leftarrow normalizar\_puntos\,(hom\,(\mathcal{u}))$
4 $\mathrm{H}'_k \leftarrow transf\_lineal\_directa\,(\mathcal{Q}n, \mathcal{u}n)$
5 ▷ Refinamiento de parámetros
6 $\mathrm{H}_k \leftarrow optimización\_numérica\left(\mathrm{H}'_k, delta\_min = 10^{-4}, max\_iteraciones = 100\right)$
7 ▷ Escalamiento
8 $\mathbf{H}_k = invertir\_normalización\left(\mathrm{H}_k, N_{\mathcal{Q}}, N_{\mathcal{u}}\right)$

A continuación se describe brevemente la acción de cada uno de los pasos del método, utilizando como guía de referencia la numeración de los renglones.

Estimación inicial lineal

• *Normalización de puntos*

A causa de la diferencia de rangos entre los puntos de referencia y sus proyecciones, los conjuntos $\mathcal{Q}$ y $\mathcal{u}$ son convertidos a *coordenadas homogéneas* considerando $s = 1$ en (2.6a). Después se normaliza cada lista de puntos de manera que el espacio 2D es acotado por un círculo unitario o por una esfera de radio uno para 3D.

De manera que un punto bidimensional $\dot{u}_j = \left[u_x, u_y\right]_j^T$, es normalizado con la siguiente expresión [41]

$$\dot{u}_{\eta_j} = hom^{-1}\left(N_{\mathcal{u}} \cdot \widetilde{\dot{u}_j}\right) = hom^{-1}\left(\begin{bmatrix} s_x & 0 & -s_x\overline{u_x} \\ 0 & s_y & -s_y\overline{u_y} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix}_j\right) \tag{2.20}$$

donde la matriz $N_{\mathcal{u}}$ realiza el dimensionamiento y sus elementos son el centro del círculo unitario y las respectivas varianzas, calculadas de la siguiente manera

$$\overline{\dot{u}} = \left[\overline{u_x}, \overline{u_y}\right]^T = \frac{1}{N}\sum_{j=1}^{N}\left(u_{x_j}, u_{y_j}\right) \tag{2.21a}$$

$$\sigma^2{}_x = \frac{1}{N}\sum_{j=1}^{N}\left(\dot{u_x}_j - \overline{\dot{u_x}}\right)^2 \tag{2.21b}$$

$$s_x = \sqrt{2/\sigma^2{}_x} \tag{2.21c}$$

$$s_y = \sqrt{2/\sigma^2{}_y} \tag{2.21d}$$

$$\sigma^2{}_y = \frac{1}{N}\sum_{j=1}^{N}\left(\dot{u_y}_j - \overline{\dot{u_y}}\right)^2 \tag{2.21e}$$

La función $normalizar\_puntos$ en la línea 2 representa este proceso descrito, que es semejante para los puntos 3D como se describe en [39].

• *Estimación inicial de la matriz de homografía*

Después de normalizar, los valores iniciales de $\mathbf{H}_k$ son obtenidos al aplicar la ***Transformada Lineal Directa***, abreviada DLT en inglés. Este método fue propuesto por Abdel-Aziz y Karara en [53], el procedimiento obtiene el valor de un conjunto de variables con base en una colección de relaciones, en este caso determinadas por la conversión (2.6b) para cada proyección $\dot{u}_j$ con base en (2.17b), obteniendo

$$u_{x_j} = \frac{u_{x_j}}{1} = \frac{\mathrm{H'}_{11}X_j + \mathrm{H'}_{12}Y_j + \mathrm{H'}_{13}Z_j + \mathrm{H'}_{14}}{\mathrm{H'}_{31}X_j + \mathrm{H'}_{32}Y_j + \mathrm{H'}_{33}Z_j + \mathrm{H'}_{34}} \tag{2.22a}$$

$$u_{y_j} = \frac{u_{y_j}}{1} = \frac{\mathrm{H'}_{21}X_j + \mathrm{H'}_{22}Y'_j + \mathrm{H'}_{23}Z_j + \mathrm{H'}_{24}}{\mathrm{H'}_{31}X_j + \mathrm{H'}_{32}Y_j + \mathrm{H'}_{33}Z_j + \mathrm{H'}_{34}} \tag{2.22b}$$

reordenando términos, ambas expresiones (2.22) se reescriben como

$$u_{x_j}X_j\mathrm{H'}_{31} + u_{x_j}Y_j\mathrm{H'}_{32} + u_{x_j}\mathrm{H'}_{33}Z_j + u_{x_j}\mathrm{H'}_{34} - \mathrm{H'}_{11}X_j - \mathrm{H'}_{12}Y_j - \mathrm{H'}_{13}Z_j - \mathrm{H'}_{14} = 0 \tag{2.23a}$$

$$u_{y_j}X_j\mathrm{H'}_{31} + u_{y_j}Y_j\mathrm{H'}_{32} + u_{y_j}\mathrm{H'}_{33}Z_j + u_{y_j}\mathrm{H'}_{34} - \mathrm{H'}_{21}X_j - \mathrm{H'}_{22}Y_j - \mathrm{H'}_{23}Z_j - \mathrm{H'}_{24} = 0 \tag{2.23b}$$

Por lo tanto para N puntos de referencia $\dot{Q}_j$, a partir del par (2.23) se determina un sistema lineal de $2N$ ecuaciones, escrito en forma matricial de la siguiente manera

$$\mathbf{L} \cdot h_k = \overline{0} \tag{2.24}$$

en donde la matriz $\mathbf{L}$ contiene todos los coeficientes que multiplican a los elementos de H'_k

$$L = \begin{bmatrix} -X_0 & -Y_0 & -Z_0 & -1 & 0 & 0 & 0 & 0 & u_{x_0}X_0 & u_{x_0}Y_0 & u_{x_0}Z_0 & u_{x_0} \\ 0 & 0 & 0 & 0 & -X_0 & -Y_0 & -Z_0 & -1 & u_{y_0}X_0 & u_{y_0}Y_0 & u_{y_0}Z_0 & u_{y_0} \\ -X_1 & -Y_1 & -Z_1 & -1 & 0 & 0 & 0 & 0 & u_{x_1}X_1 & u_{x_1}Y_1 & u_{x_1}Z_1 & u_{x_1} \\ 0 & 0 & 0 & 0 & -X_1 & -Y_1 & -Z_1 & -1 & u_{y_1}X_1 & u_{y_1}Y_1 & u_{y_1}Z_1 & u_{y_1} \\ & & & & & \vdots & & & & & & \\ -X_{N-1} & -Y_{N-1} & -Z_{N-1} & -1 & 0 & 0 & 0 & 0 & u_{x_{N-1}}X_{N-1} & u_{x_{N-1}}Y_{N-1} & u_{x_{N-1}}Z_{N-1} & u_{x_{N-1}} \\ 0 & 0 & & 0 & -X_{N-1} & -Y_{N-1} & -Z_{N-1} & -1 & u_{y_{N-1}}X_{N-1} & u_{y_{N-1}}Y_{N-1} & u_{y_{N-1}}Z_{N-1} & u_{y_{N-1}} \end{bmatrix}$$

y h_k es un vector definido por las variables incógnita, es decir los elementos de la matriz H'_k de acuerdo al orden de los factores de $\mathbf{L}$ y al par de expresiones (2.23)

$$h_k = \begin{bmatrix} \mathrm{H}'_{11} & \mathrm{H}'_{12} & \mathrm{H}'_{13} & \mathrm{H}'_{14} & \mathrm{H}'_{21} & \mathrm{H}'_{22} & \mathrm{H}'_{23} & \mathrm{H}'_{24} & \mathrm{H}'_{31} & \mathrm{H}'_{32} & \mathrm{H}'_{33} & \mathrm{H}'_{34} \end{bmatrix}^T$$

Entonces al contar con 12 incógnitas, seis puntos $\dot{Q}_j$ y sus respectivas proyecciones $\dot{u}_j$ son suficientes para resolver el sistema (2.24), y con ello obtener la estimación inicial de $\mathbf{H}_k$. Al estructurarse en *coordenadas homogéneas*, la *matriz de homografía* suele ser determinada al factorizarse con el método de *Descomposición en Valores Singulares* SVD por sus siglas en inglés, el procedimiento de esta técnica aplicado para H'_k es descrito en [39, 41].

Refinamiento de parámetros

La *matriz de homografía* calculada por DLT, en la mayoría de los casos no define adecuadamente la correspondencia 3D→2D, porque la relación en coordenadas cartesianas no es lineal. Por lo tanto la proyección de un punto de referencia $\dot{Q}_j$, utilizando H'_k es una aproximación a la verdadera posición de $\dot{u}_j$ que se indica en la línea 4 del Algoritmo 2.2.1.

Esta diferencia es cuantificada para las N proyecciones estimadas $\dot{u}'_j$ de una imagen, por medio del ***error cuadrático de reproyección*** definido para una imagen k como

$$\mathrm{E}_{proy_k} = \sum_{j=0}^{N-1} \left\| \dot{u}_j - \dot{u}'_j \right\|^2 = \sum_{j=0}^{N-1} \left\| \dot{u}_j - hom^{-1}\left(\mathrm{H}'_k \cdot \widetilde{\dot{Q}_j}\right) \right\|^2 \tag{2.25}$$

para N puntos de referencia detectados en una imagen k del *patrón de calibración* [39]. Para solucionar esto, se aplica un método iterativo de *optimización numérica* a la matriz H'_k, de manera que al estimar una nueva homografía H''_k, el *error de proyección* (2.25) se reduzca cada vez más hasta llegar a un valor deseado o realizar un determinado número de iteraciones, este procedimiento es indicado en la línea 6 [42].

El método de optimización más utilizado para este caso es el *algoritmo de Levenberg-Marquart*, abreviado como LM en inglés [54, 55]. El procedimiento desarrollado y aplicado directamente a la matriz de homografía H'_k, es especificado paso por paso en [41].

Escalamiento

Finalmente para obtener la *matriz de homografía* del modelo de proyección (2.17), se debe volver a dimensionar los elementos de H_k calculados en la etapa de refinamiento. Dicho escalamiento se define mediante las matrices de normalización por la siguiente expresión [39, 41]

$$\mathbf{H}_k = {N_u}^{-1} \cdot \mathrm{H}_k \cdot N_Q \tag{2.26}$$

Con esta etapa de escalamiento concluye el Algoritmo 2.2.1. Después en función de la precisión que se requiera para la aplicación, se debe seleccionar el método más adecuado y consecuentemente el tipo de referencia 3D a utilizar.

Algunos métodos de calibración hacen cambios menores al procedimiento presentado para obtener la *matriz de homografía*, por la cantidad de diferentes perspectivas del *patrón de calibración* que necesiten y/o considerar más coeficientes de distorsión óptica. A continuación se describirá la metodología elegida para calibrar las cámaras usadas en este proyecto.

2.2.2 Calibración por observación de planos

Es una metodología que infiere los parámetros intrínsecos y coeficientes de distorsión óptica, utilizando como guía de referencia escenas con estructuras que contienen o forman rectas paralelas y ortogonales, de longitud conocida definidas en uno o más planos.

Esta técnica propuesta inicialmente por Tsai en [52], ha evolucionado de diferentes maneras conservando el principio de no conocer la posición del *patrón de calibración* y utilizar solo mediciones métricas y de pixeles en las imágenes. Esta característica lo convirtió en tendencia dentro del área de visión por computadora, para aplicaciones que no requieren de alta precisión.

Particularmente para los propósitos de este trabajo, se necesitan obtener los parámetros de las cámaras IR y VIS ocupando el mismo objeto de referencia, para determinar la correspondencia de coordenadas entre ambas imágenes.

El *patrón de calibración* conocido como *tablero de ajedrez*, se ha utilizado como referencia compartida de cámaras IR y VIS en [56–59], demostrando ser funcional con el *método de calibración por observación de un plano* propuesto por Bouguet [60], del cual a continuación se presenta una síntesis del procedimiento considerando la documentación [39, 41, 50, 60, 61].

Algoritmo 2.2.2: Calibración monocular usando múltiples perspectivas de un plano.—*Parte I*—

Entrada(s): $\mathcal{Q} = \left[\dot{Q}_1, \dot{Q}_2, \dots, \dot{Q}_j, \dots, \dot{Q}_N\right] \Rightarrow N$-puntos de referencia, donde $\dot{Q}_j = \left[X_j, Y_j, 0\right]^T$

$\mathcal{U} = [\mathcal{u}_1, \mathcal{u}_2, \dots, \mathcal{u}_m, \dots, \mathcal{u}_k] \Rightarrow k$-conjuntos de puntos de proyección de referencia, donde

$\mathcal{u} = \left[\dot{u}_1, \dot{u}_2, \dots, \dot{u}_j, \dots, \dot{u}_N\right] \Rightarrow$ Lista de N-proyecciones de los respectivos puntos de referencia

cada uno definido como $\dot{u}_j = \left[u_{x_j}, u_{y_j}\right]^T$

Salida(s): Matriz intrínseca K (2.16); Coeficientes de distorsión $\mathfrak{d}_{rc}$ (2.14)

▷ Cálculo de matrices de homografía
para cada $\mathcal{u}_m$ ***de*** $\mathcal{U}$ **hacer**
  $\mathbf{H}_m \leftarrow calcular\_homografía\,(\mathcal{u}_m, \mathcal{Q})$ ▷ Algoritmo 2.2.1
  $\mathcal{H} \leftarrow adjuntar\,\{\mathbf{H}_m\}$
fin
▷ Estimación inicial de parámetros intrínsecos
$\mathrm{K}_0 \leftarrow intrínsecos\_iniciales\,(\mathcal{H}, img\_ancho = 320, img\_alto = 240)$ ▷ $\mathrm{K}_0 = \begin{bmatrix} \alpha & 0 & c_x \\ 0 & \beta & c_y \\ 0 & 0 & 1 \end{bmatrix}$
▷ Estimación inicial de parámetros extrínsecos
para cada $\mathbf{H}_m$ ***de*** $\mathcal{H}$ **hacer**
  $\mathbf{R}_{3_m}, t_m \leftarrow extrinsecos\_iniciales\,(\mathbf{H}_m, \mathrm{K}_0)$
  $\mathcal{W} \leftarrow adjuntar\,\{\mathbf{R}_3\mathbf{T}_m\}$
fin

Algoritmo 2.2.2: Calibración monocular usando múltiples perspectivas de un plano.–*Parte II*–

13 ▷ Refinamiento de parámetros

14 $\mathrm{K}, \mathfrak{d}_{rc} \leftarrow optimización\_numérica\left(\mathrm{K}_0, \mathcal{W}, \mathcal{Q}, \mathcal{U}, delta\_min = 10^{-4}, max\_iteraciones = 100\right)$

Utilizando como referencia la numeración de renglones del Algoritmo 2.2.2, a continuación se describe brevemente cada parte del procedimiento.

Cálculo de matrices de homografía

Al utilizar un *patrón de calibración* bidimensional, todos los puntos $\dot{Q}_j$ están posicionados en el plano definido por Z_j, de tal manera que al considerar que $Z_j = 0$ el modelo de proyección (2.19) se simplifica de la siguiente manera

$$\begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix}_j = \begin{bmatrix} H_{11} & H_{12} & H_{13} & H_{14} \\ H_{21} & H_{22} & H_{23} & H_{24} \\ H_{31} & H_{32} & H_{33} & H_{34} \end{bmatrix}_k \cdot \begin{bmatrix} X_j \\ Y_j \\ 0 \\ 1 \end{bmatrix} \equiv \begin{bmatrix} H_{11} & H_{12} & H_{14} \\ H_{21} & H_{22} & H_{24} \\ H_{31} & H_{32} & H_{34} \end{bmatrix}_k \cdot \begin{bmatrix} X_j \\ Y_j \\ 1 \end{bmatrix} \tag{2.27}$$

En consecuencia el sistema de ecuaciones (2.24) se reduce a nueve incógnitas por resolver, porque todos los términos multiplicados por la componente Z_j en (2.23) son descartados simplificando su solución con el Algoritmo 2.2.1, como se indica en el renglón 3.

Cada una de las matrices que se obtiene de la función $calcular\_homografía$, se adjunta a la lista $\mathcal{H}$ para las etapas posteriores. El desarrollo del cálculo de esta fase, puede ser consultado en [41, 50].

Estimación inicial de parámetros intrínsecos

Los valores iniciales de los elementos de la matriz (2.16), característicos de la cámara son definidos de la siguiente manera:

1. La *distorsión diagonal* γ se considera nula $\gamma = 0$, porque los sensores de las cámaras digitales no presentan alguna curvatura de sus ejes coordenados.
2. El centro $C_s = \left[c_x, c_y\right]^T$ es igual al punto de simetría del *plano de la imagen*, medido por la cantidad de pixeles que la forman a lo largo y ancho, es decir

   $$C_s = 0.5 \cdot \left[img.max\_u_x, img.max\_u_y\right]^T$$

   en donde $img.max$ señala el máximo valor de la abscisa y ordenada en la imagen.
3. Las *distancias focales* α y β, se determinan al solucionar un sistema de ecuaciones, que se plantea por la proyección de *puntos de fuga* ortogonales denominados en inglés como ***vanishing points***, dicho procedimiento es descrito en [61] utilizando las matrices la lista $\mathcal{H}$.

Este procedimiento descrito esta representado por la función $intr\acute{i}nsecos\_iniciales$, señalado en la línea 7 del Algoritmo 2.2.2.

Estimación inicial de parámetros extrínsecos

Al conocer la matriz K_0, se pueden calcular los *parámetros extrínsecos* (2.2) y (2.4) preliminares de cada conjunto $\mathcal{u}_m$, descomponiendo su respectiva $\mathbf{H}_m$ de la siguiente manera

$$\begin{bmatrix} H_{11} & H_{12} & H_{14} \\ H_{21} & H_{22} & H_{24} \\ H_{31} & H_{32} & H_{34} \end{bmatrix}_m \equiv \mathrm{K}_0 \cdot \begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ r_{31} & r_{32} & t_z \end{bmatrix}_m \Leftrightarrow \begin{bmatrix} h_{k_1} & h_{k_2} & h_{k_3} \end{bmatrix} \equiv \mathrm{K}_0 \cdot \begin{bmatrix} r_{k_1} & r_{k_2} & t_k \end{bmatrix} \qquad (2.28)$$

de donde al expresar las matrices con vectores por cada columna, se obtienen las correspondencias

$$r_{k_1} = \lambda \cdot {\mathrm{K}_0}^{-1} \cdot h_{k_1}; \qquad r_{k_2} = \lambda \cdot {\mathrm{K}_0}^{-1} \cdot h_{k_2}; \qquad r_{k_3} = r_{k_1} \times r_{k_2}; \qquad t_k = \lambda \cdot {\mathrm{K}_0}^{-1} \cdot h_{k_4}$$

y

$$\lambda = \frac{1}{\left\| {\mathrm{K}_0}^{-1} \cdot h_{k_1} \right\|} = \frac{1}{\left\| {\mathrm{K}_0}^{-1} \cdot h_{k_2} \right\|}$$

Los vectores columna r_{k_1}, r_{k_2} y r_{k_3}, son determinados por la representación de Euler-Rodriguez [39], por lo que para obtener (2.2) es necesario hacer la conversión desarrollada en [41], para que posteriormente se calcule la matriz extrínseca (2.10) y pueda ser adjuntada a la lista $\mathcal{W}$.

Refinamiento de parámetros

Los valores iniciales de la matriz intrínseca y de cada conjunto de parámetros extrínsecos, definen una aproximación de la proyección de los puntos $\dot{u}_{m,j}$. La diferencia entre la estimada y real solo puede disminuirse al modificar los valores iniciales de las etapas previas, con un método iterativo de *optimización numérica*. Este procedimiento es semejante al aplicado en la última etapa del Algoritmo 2.2.1, pero en esta ocasión el ***error cuadrático de reproyección*** se calcula con la siguiente expresión

$$\mathrm{E}_{proy_T} = \sum_{m=0}^{k-1} \sum_{j=0}^{N-1} \left\| \dot{u}_{m,j} - \dot{u}'_{m,j} \right\|^2 \qquad (2.29)$$

definiendo cada proyección estimada de un punto de referencia $\dot{Q}_j$ como

$$\dot{u}'_{m,j} = hom^{-1}\left(\mathrm{K}_0 \cdot \widetilde{q_{\dot{m},j_{sd}}} \right) \qquad (2.30a)$$

en donde $\widetilde{q_{\dot{m},j_{sd}}}$ es la proyección (2.13a) que incluye distorsión *radial* y *tangencial*, convertida en *coordenadas homogéneas* por (2.6a), y los parámetros extrínsecos se incluyen en el modelo normalizado (2.9b)

$$\dot{q}_{m,j} = hom^{-1}\left(\mathbf{R_3 T}_m \cdot \widetilde{\dot{Q}_j} \right) \qquad (2.30b)$$

Los coeficientes (2.14) son inicialmente considerados cero, y estos se van obteniendo con cada iteración del método de *Levenberg-Marquart* LM. Como guía de referencia sobre la aplicación del procedimiento, se puede utilizar el desarrollo presentado en [41], con la reserva de considerar

solo (2.11) en el modelo de proyección estimada.

Únicamente los *parámetros intrínsecos* y *coeficientes de distorsión* son optimizados, porque sus valores son independientes de la posición de los puntos con respecto de la cámara. Por esta razón al utilizarlos con la *transformación del cuerpo rígido*, es posible relacionar dos perspectivas diferentes, de una misma escena por medio de geometría como se describirá en la siguiente Sección.

2.3 Correspondencia geométrica estereoscópica

El modelo de *cámara finita* presentado en la Sección 2.1, en conjunto con la definición de sus *parámetros intrínsecos* y *coeficientes de distorsión óptica* descritos en la Sección 2.2, relaciona los puntos de sus imágenes con magnitudes métricas de los objetos proyectados, habilitando un filtro adicional en algunos detectores de objetos, como en [7].

Considerando como fundamento los conceptos anteriores, en esta Sección se introducirán los principios de la geometría de proyección, utilizando dos cámaras con un *campo de visión* compartido, y posteriormente se abordará la correspondencia de puntos entre las imágenes IR y VIS.

2.3.1 Geometría de proyección epipolar

Un punto $\dot{Q}_j$ 3D al proyectarse en una imagen, se ubica a cierta distancia de la cámara sobre una recta, que representa el rayo que pasa por C_p e incide en el *plano de la imagen* $\mathrm{I}_s{}^{-1}$, definiendo su proyección $\dot{q}_{1,j}$, como se observa en la Figura 2.6. Bajo esta suposición, sí una segunda cámara observa el mismo punto de la escena, su proyección $\dot{q}_{2,j}$ en la segunda imagen, también se ubica en otra recta que viene de $\dot{Q}_j$.

El rayo que proyecta el punto $\dot{q}_{1,j}$ en la imagen de la segunda cámara, se proyecta como una recta denominada ***línea epipolar***, la cual representa el espacio de la posible ubicación del punto $\dot{Q}_j$. Esta afirmación también aplica para el caso inverso, de la cámara dos a la cámara uno, definiendo una relación geométrica conocida como ***restricción epipolar***, que indica lo siguiente:

> *Para cualquier punto $\dot{q}_{1,j}$ en la primera imagen I_{s1}, su punto correspondiente en la segunda imagen I_{s2} está restringido a ubicarse sobre una línea, dependiendo únicamente de los **parámetros intrínsecos** de ambas cámaras, así como de su traslación y rotación relativa entre ellas [40].*

La condición citada en el párrafo anterior, es ilustrada en la Figura 2.6. Para múltiples puntos, la restricción define una *línea epipolar* por cada punto proyectado en la imagen uno I_{s1}, sobre la imagen dos I_{s2} y viceversa.

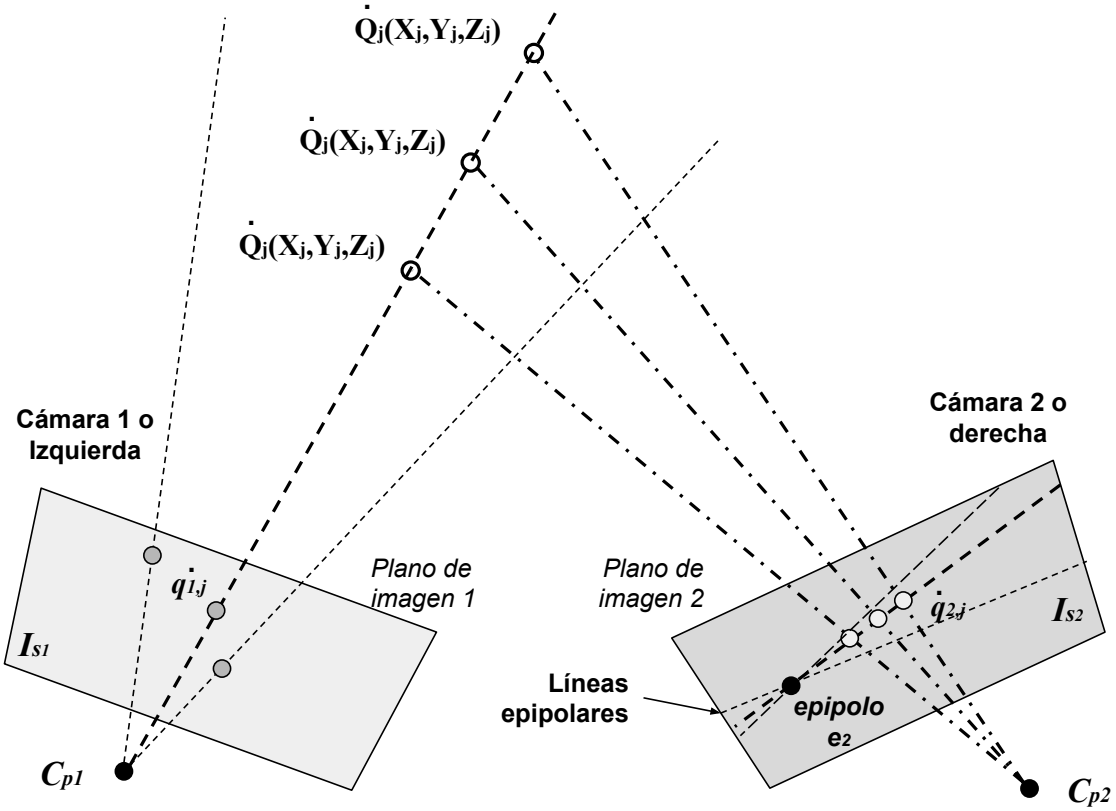


Figura 2.6. Restricción epipolar descrita de forma gráfica.

Todas las *líneas epipolares* de una imagen, convergen en un punto, denominado ***epipolo*** que puede ubicarse en la recta que une los centros ópticos de las cámaras, en la imagen opuesta o fuera del plano de la imagen, porque depende de la posición de las cámaras [39].

Estas relaciones geométricas de proyección, son la base del modelo de correspondencia de coordenadas entre dos imágenes, cuya descripción se desarrolla a continuación.

2.3.2 Asociación por proyección normalizada

Es la *correspondencia geométrica* que relaciona dos proyecciones diferentes de un punto de referencia, observado por dos cámaras cuya posición espacial y parámetros intrínsecos determinan la asociación.

De forma especifica al tener dos cámaras visualizando un punto $\dot{Q}_j$ 3D dentro de su campo de visión compartido, las dos proyecciones se definen de acuerdo al sistema de referencia de cada equipo considerando el modelo (2.9a), como se muestra a continuación

$$s_1\widetilde{q_{1,j}} = \mathbf{R_3T}_1 \cdot \widetilde{\dot{Q}_j} = M_p\big|_{f=1} hom\left(R_{3_1} \cdot \dot{Q}_j + t_1\right) \equiv R_{3_1} \cdot \dot{Q}_j + t_1 \tag{2.31a}$$

$$s_2\widetilde{q_{2,j}} = \mathbf{R_3T}_2 \cdot \widetilde{\dot{Q}_j} = M_p\big|_{f=1} hom\left(R_{3_2} \cdot \dot{Q}_j + t_2\right) \equiv R_{3_2} \cdot \dot{Q}_j + t_2 \tag{2.31b}$$

en donde s_1 y s_2 son los respectivos factores de escalamiento de acuerdo a la definición (2.6a). Entonces para obtener la correspondencia se selecciona un sistema de coordenadas de las cámaras, por ejemplo el marco de referencia de la cámara uno, por lo cual (2.31a) se reescribe como $s_1\widetilde{q_{1,j}} = \dot{Q}_j$ y esta expresión puede sustituirse en (2.31b) por la equivalencia (2.6c) obteniendo

$$s_2\widetilde{q_{2,j}} = s_1\mathbf{R_3}_2 \cdot \widetilde{q_{1,j}} + t_2 \tag{2.32}$$

esta expresión restringe la posición de las proyecciones $q_{1,j}$ y $q_{2,j}$, equiparable a la condición definida por *líneas epipolares*. Es decir, que (2.32) define la posición de los puntos de la cámara dos, con respecto al sistema de referencia de la cámara uno. Esta relación convencionalmente se define por una matriz, que se obtiene al pre-multiplicar (2.32) por el vector $\tilde{t}_2$ con un producto cruz de ambos lados

$$s_2 t_2 \times \widetilde{q_{2,j}} = s_1 t_2 \times \mathbf{R_3}_2 \cdot \widetilde{q_{1,j}} + t_2 \times t_2 \tag{2.33a}$$

donde el producto $t_2 \times t_2 = 0$. Después (2.33a) se multiplica por $\widetilde{q_{2,j}}$ en ambos lados con un producto punto, haciendo que el término a la izquierda del signo igual sea cero, porque el punto es ortogonal al vector de traslación t_2, obteniendo

$$s_1\widetilde{q_{2,j}}^T \cdot t_2 \times \mathbf{R_3}_2 \cdot \widetilde{q_{1,j}} = s_1\widetilde{q_{2,j}}^T \cdot [t_2]_\times \cdot \mathbf{R_3}_2 \cdot \widetilde{q_{1,j}} = 0 \tag{2.33b}$$

$[t_2]_\times$ es la matriz de conversión utilizada para operar con el vector de traslación (2.4), de modo que el producto cruz es equivalente a multiplicar por este término

$$[t_2]_\times = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \tag{2.33c}$$

y al considerar que los factores de escala s_1 y s_2 son unitarios, el producto $[t_2]_x \cdot \mathbf{R_3}_2$ de (2.33b), define la correspondencia de proyección entre los puntos, razón por la que es expresada por un solo término de la siguiente manera

$$\widetilde{q_{2,j}}^T \cdot \mathbf{E} \cdot \widetilde{q_{1,j}} = 0 \tag{2.34}$$

en donde $\mathbf{E}$ es denominada como ***matriz esencial*** y determina la correlación geométrica de los puntos de proyección entre dos cámaras, que comparten un mismo campo de visión [40].

Esta relación vincula las proyecciones normalizadas de los puntos, en *coordenada homogéneas* entre dos *planos de imagen*, la inclusión de los *parámetros intrínsecos* de cada cámara para definir la asociación a nivel pixel se describe a continuación.

2.3.3 Asociación por proyección específica

Esta correspondencia se deduce de un procedimiento semejante al expuesto para obtener la *matriz esencial*, con la diferencia de considerar que las proyecciones de puntos de cada cámara, ahora son determinadas con el modelo (2.17a) el cual se puede factorizar considerando (2.15) de la siguiente manera

$$s_1\widetilde{q_{1,j}} = \mathrm{K}_1\mathbf{R_3}\mathbf{T}_1 \cdot \widetilde{Q_j} = \mathrm{K}_1 \cdot M_{p_0} \cdot hom\left(R_{3_1} \cdot \dot{Q}_j + t_1\right) \equiv \mathrm{K}_1\left(R_{3_1} \cdot \dot{Q}_j + t_1\right) \tag{2.35a}$$

$$s_2\widetilde{q_{2,j}} = \mathrm{K}_3\mathbf{R_3}\mathbf{T}_2 \cdot \widetilde{Q_j} = \mathrm{K}_2 \cdot M_{p_0} \cdot hom\left(R_{3_2} \cdot \dot{Q}_j + t_2\right) \equiv \mathrm{K}_2\left(R_{3_2} \cdot \dot{Q}_j + t_2\right) \tag{2.35b}$$

a partir de estas expresiones se desarrolla el mismo procedimiento algebraico del que se obtuvo (2.33b), pero en esta ocasión el proceso involucra a las matrices intrínsecas de cada cámara, obteniendo la siguiente expresión

$$\widetilde{\dot{q}_{2,j}}^{T} \cdot \mathrm{K_2}^{-T} \cdot \mathbf{E} \cdot \mathrm{K_1}^{-1} \cdot \widetilde{\dot{q}_{1,j}} = 0 \tag{2.36}$$

$\mathrm{K}^{-T} = \mathrm{K}^{-1^{T}}$ indica la *matriz intrínseca* inversa y después transpuesta. (2.36) es la correspondencia de pixeles en la imagen de la cámara dos a la uno, la cual es representada por el término $\mathbf{F}$ como

$$\widetilde{\dot{q}_{2,j}}^{T} \cdot \mathbf{F} \cdot \widetilde{\dot{q}_{1,j}} = 0 \tag{2.37}$$

en donde $\mathbf{F} = \mathrm{K_2}^{-T} \cdot \mathbf{E} \cdot \mathrm{K_1}^{-1}$, se le denomina como ***matriz fundamental*** [39]. Una forma de calcular $\mathbf{F}$ es aplicando el *algoritmo de ocho puntos* propuesto por Higging [62], el cual es descrito paso por paso en [40]. Al conocer los *parámetros intrínsecos* de cada cámara y la *matriz fundamental*, es posible calcular la *matriz esencial* por medio de la siguiente relación

$$\mathbf{E} = \mathrm{K_2}^{T} \cdot \mathbf{F} \cdot \mathrm{K_1} \tag{2.38}$$

Al descomponer la matriz $\mathbf{E}$ calculada por (2.38) por el método de SVD, se obtiene la matriz de rotación $\mathbf{R_E}$ y el vector de traslación $\mathbf{t_E}$, que establecen la correspondencia de coordenadas de la cámara uno a la cámara dos como

$$\widetilde{\dot{q}_{1,j}} = \mathbf{R_E} \cdot \widetilde{\dot{q}_{2,j}} + \mathbf{t_E} \tag{2.39}$$

Esta relación ilustrada en la Figura 2.7, finalmente proporciona la correspondencia entre los puntos de la cámara.

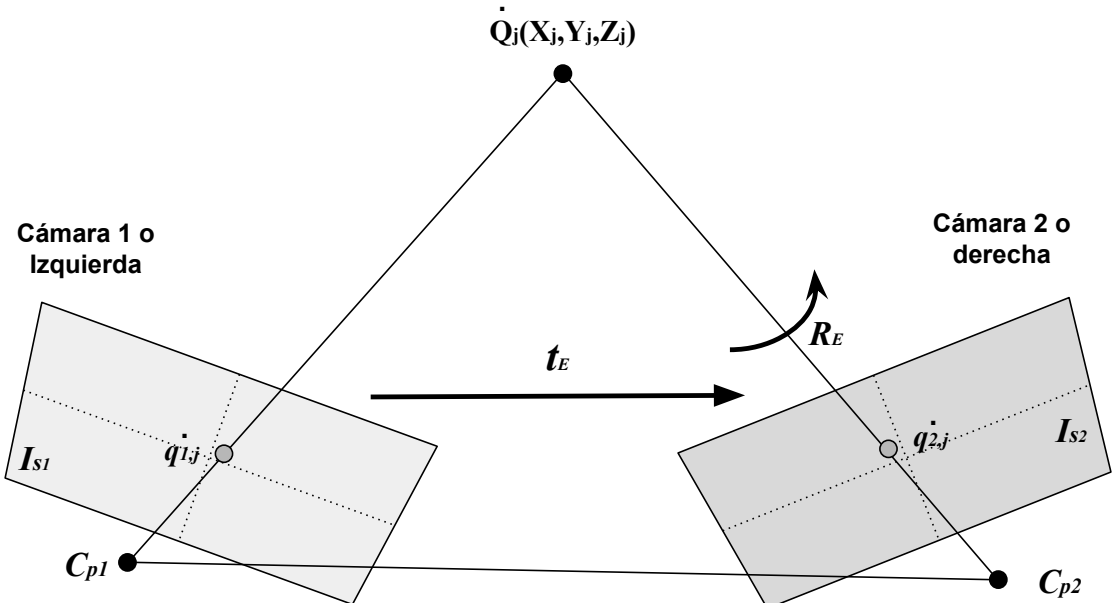


Figura 2.7. Relación de puntos en la imagen en pixeles, entre dos cámaras.

2.4 Métodos alternos de correspondencia

Además de la metodología presentada la Sección 2.3, para relacionar los puntos de cada imagen de dos cámaras usando *geometría de proyección*, durante la documentación previa para realizar el proyecto, se encontraron otros planteamientos para poder obtener dicha correspondencia, por lo que a continuación se describirán brevemente dos de ellos.

2.4.1 Interpolación geométrica

Este método es descrito en [56], desarrollado como parte de un sistema para detectar peatones a la intemperie bajo diferentes condiciones de iluminación. El detector ocupa las imágenes de dos cámaras; una cámara web VIS y una cámara IR sensible a LWIR, ambas posicionadas para tener un campo de visión compartido y alineando las lentes sobre la horizontal a nivel macroscópico.

Para definir la correspondencia de puntos, utilizan un *patrón de calibración* formado por una malla de líneas paralelas y perpendiculares, cuyas intersecciones definen 24 puntos de referencia 2D $\dot{Q}_j$. De estas marcas, cuatro son utilizadas para definir la *matriz de transformación geométrica*, a partir del siguiente sistema de ecuaciones

$$\begin{bmatrix} r_{x_1} & r_{x_2} & r_{x_3} & r_{x_4} \\ r_{y_1} & r_{y_2} & r_{y_3} & r_{y_4} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} u_{x_1} & u_{x_2} & u_{x_3} & u_{x_4} \\ u_{y_1} & u_{y_2} & u_{y_3} & u_{y_4} \\ u_{x_1}u_{y_1} & u_{x_2}u_{y_2} & u_{x_3}u_{y_3} & u_{x_4}u_{y_4} \\ 1 & 1 & 1 & 1 \end{bmatrix} \tag{2.40}$$

en donde $\dot{r}_j = [r_{x_j}, r_{y_j}]^T$ es la proyección del punto de referencia $\dot{Q}_j$, en la imagen de la cámara IR y $\dot{u}_j = [u_{x_j}, u_{y_j}]^T$ es la proyección de $\dot{Q}_j$ pero en la cámara VIS. Después de resolver el sistema (2.40), la correspondencia punto a punto entre ambas imágenes la definen como

$$\begin{bmatrix} r_{x_j} \\ r_{y_j} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} u_{x_1} \\ u_{y_1} \\ u_{x_1}u_{y_1} \\ 1 \end{bmatrix} \tag{2.41}$$

Con los 20 puntos de referencia restantes $\dot{Q}'_j$, midieron el error cuadrático medio de reproyección calculando las estimaciones de sus proyecciones en la imagen IR, usando la correspondencia (2.41) y las proyecciones definidas inicialmente en cada imagen. En [56] reportaron un error de correlación de un pixel, además de indicar que utilizan este modelo en otros trabajos del mismo autor, especificando como montaron las cámaras en una base y resaltando que solo es necesario realizar una vez este procedimiento.

2.4.2 Semejanza de contraste y gradientes

Este método busca la correspondencia de *bounding boxes* entre dos imágenes tipo IR y VIS, que capturan una misma escena dentro de su campo de visión compartido. La relación entre ambas cámaras, se determina ponderando la similitud de las regiones seleccionadas en cada imagen.

El procedimiento comienza por normalizar el rango dinámico de ambas imágenes en escala de grises. Después define un cuadrilítero en alguna de las dos imágenes, por ejemplo en la VIS, como un vector cuyos elementos son; el ancho y alto de la región, seguidos de las coordenadas del punto central de la ventana. Posteriormente, en la imagen IR se definen un conjunto de *bounding boxes*, de dimensiones variables en función de un registro de profundidad y con base en el tamaño de la región definida en la imagen VIS.

La comparación de las regiones de imagen se hace tomando en cuenta estos dos factores:

1. Una función de similitud denominada *mutua información*, que compara el comportamiento estadístico del contraste, en ambas regiones de las imágenes.
2. Y un ponderador que equipara la *diferencia de fases* entre los mapas de *gradientes Gaussianos* de las dos ventanas analizadas, considerando un índice de sesgo para disparidades de magnitud extraordinaria, en relación con los demás.

Este algoritmo fue propuesto en [57], como parte de un sistema que combinaba la información de una cámara esteroscópica con una infrarroja tipo LWIR. El planteamiento descrito, implementa conceptos del campo de investigación, que analiza la mezcla de imágenes IR con VIS nombrado en la literatura como *fusión espectral*.

Algunas de las aplicaciones de esta área son; la detección y seguimiento de objetos, representaciones de pseudo color y aplicaciones de vídeo vigilancia, que aprovechan las capacidades de cada cámara partiendo desde la selección de un bounding box hasta la formación de una nueva imagen de texturas combinadas de las dos fuentes. En [63, 64] se describen brevemente los diferentes métodos reportados en la literatura, en conjunto con sus aplicaciones.

Resumen

La estructuración de los temas presentados en el Capítulo 2, es una síntesis de los conceptos involucrados en la correspondencia geométrica de puntos, en las imágenes adquiridas por las cámaras IR y VIS, que se utilizarán en el detector de víctimas propuesto.

El *modelo de proyección* de la *cámara finita* (2.17), presentado en la Sección 2.1, describe el proceso de representación 3D → 2D, determinado por la transformación de movimiento de un cuerpo rígido (2.1), la correspondencia de proyección (2.8), la corrección de distorsión óptica (2.13) y por los parámetros físicos (2.16) que caracterizan a la imagen.

Los factores constantes en las proyecciones de una cámara monocular, se infieren a partir de referencias conocidas del espacio 3D, aplicando alguna metodología de *calibración*, tema central de la Sección 2.2. Este proceso comienza con la estimación de *matrices de homografía* (2.18), que representa a los parámetros *intrínsecos* y *extrínsecos* de la cámara, por eso se presentó la técnica estándar para calcularla [39].

Después, en virtud de la diferencia de imágenes a trabajar, se describió el algoritmo de *calibración por observación de múltiples perspectivas de un paralelogramo* [60], el cual fue seleccionado para usarse en el proyecto, porque utiliza un *patrón de calibración* distinguible para ambas cámaras.

Considerando como fundamento los conceptos previos, en la Sección 2.3 se presento una síntesis de la *correspondencia esteroscópica* de puntos observados por ambas cámaras, descrita por medio de *geometría epipolar* a partir de la *matriz fundamental* y *esencial* que establecen una relación de puntos entre dos imágenes, aplicando los parámetros intrínsecos de cada cámara.

Y finalmente, la Sección 2.4 se dedicó a describir dos técnicas alternas a la geométrica, para establecer la correspondencia de coordenadas. El objetivo de las reseñas que constituyen este apartado, es implementar una y comparar sus resultados con un mismo conjunto de imágenes.

Capítulo 3

Descripción de formas humanas

Las aplicaciones de visión artificial relacionadas a tareas de clasificación, detección, segmentación o rastreo de objetos, comparten en común procesos en su desarrollo e implementación, a pesar de tener diferentes fines específicos.

Uno de estos factores es la generación de patrones, que permitan reconocer los objetivos de interés, inmersos en el entorno de la escena capturada. Frecuentemente en este procedimiento, se transforma la imagen de un formato de visualización, a otros tipos de representaciones que destacan rasgos distintivos, que tienden a favorecer la creación de los modelos mencionados.

Por esta razón, en este Capítulo se presenta una breve explicación de las metodologías empleadas, para extraer y describir conjuntos de características, que se han reportado como adecuados en trabajos antecedentes, para formar *patrones de personas* utilizando imágenes de escenarios con circunstancias semejantes a las consideradas en la Sección 1.2.

Los temas de este Capítulo están organizados de la siguiente manera; en primera instancia se plantea el concepto de *características* de una imagen, seguido de la operación base utilizada para extraerlas. Posteriormente, en la Sección 3.2 se abordan los operadores matemáticos, que detectan diferentes *singularidades* de formas antropomórficas. Y en la Sección 3.3, se presentan los métodos conocidos como *descriptores de características*, que definen los elementos base de los modelos de personas.

3.1 Características de una imagen

Son *representaciones singulares de información distintiva*, que describen el contenido de una imagen o región de ésta, en un dominio distinto a los espacios de visualización gráfica. Los métodos de extracción de puntos característicos, se definen por funciones matemáticas que operan un pixel o por un conjunto de pixeles vecinos al punto de cálculo.

El principal propósito de las características es obtener *conjuntos de datos relativamente constantes para formar patrones*, que puedan ser buscados en otras imágenes semejantes con deformaciones o cambios como de iluminación, escala o rotación. En consecuencia los puntos singulares deben ser invariantes a factores inherentes de la aplicación.

Un modelo de comparación representativo de un objeto, puede ser determinado por un ***descriptor*** que define una metodología para localizar puntos singulares, utilizando ***detectores*** que forman y estructuran el patrón, mediante el análisis de un grupo de imágenes que contienen al objetivo de interés.

Los *detectores* consisten generalmente de diversos *filtros*, que se aplican a las imágenes mediante la operación ***convolución discreta*** 2D [65], que determina el valor de cada nuevo pixel $\mathrm{I}_w\,(i,j)$, como una suma ponderada de los pixeles vecinos que rodean su posición en una región cuadrada $f \epsilon \mathbb{R}^{M\times N} | M = N \epsilon \mathbb{N}$ de la imagen $\mathrm{I} \epsilon \mathbb{R}^{W\times H} | W, H \epsilon \mathbb{N}$ como

$$\mathrm{I}_w\,(i,j) = \sum_{m=1}^{m=M} \sum_{n=1}^{n=N} \underbrace{\mathrm{I}\,(i-m, j-n)}_{f(m,n)} \cdot w\,(m,n) = f * w \tag{3.1}$$

en donde w es denominado como el ***núcleo del filtro***, y contiene los coeficientes que definen su efecto. Ambos operadores f y w deben tener el mismo tamaño impar, considerando como mínimo matrices de 3×3 elementos.

La imagen filtrada I_w es resultado de aplicar la operación (3.1) iterativamente, desplazando la estructura cuadrada que define el vecindario f por el espacio de I. El movimiento de la región se realiza con un determinado paso $\mathbb{s}$, que puede modificar las dimensiones de la imagen de la siguiente manera

$$W_{\mathrm{I}_w} = \frac{W_{\mathrm{I}} + 2\mathbb{p} - M_w}{\mathbb{s}} + 1 \tag{3.2a}$$

$$H_{\mathrm{I}_w} = \frac{H_{\mathrm{I}} + 2\mathbb{p} - N_w}{\mathbb{s}} + 1 \tag{3.2b}$$

Para no submuestrear la imagen y conservar su tamaño original, I se extiende agregando $\mathbb{p}$ pixeles en sus extremos, habilitando la operación en todos los elementos que integran el perímetro. El valor de dichos pixeles es determinado por diversas técnicas conocidas como *relleno o* ***padding***, algunos de estos métodos pueden ser consultados en [42, 65, 66] así como información complementaria sobre la convolución.

3.2 Detectores de características

En esta Sección se presentan las funciones que definen los núcleos de los filtros, implementados en los descriptores más utilizados para la detección de personas. Los extractores de *puntos característicos* de interés para este trabajo, consisten principalmente de *operadores diferenciales* que definen bordes y localizan singularidades en el espacio escala.

Además de los filtros que se explicarán a continuación, en la implementación del trabajo también se emplearon filtros de preproceso para fines de obtener mejores modelos de referencia para el algoritmo clasificador, por esta razón serán mencionados en su respectivo momento.

3.2.1 Filtro gaussiano

La presencia de ruido ocasionado por el sensor de la cámara y/o la cantidad de detalles en las imágenes de trabajo, dificultan la extracción de puntos característicos, sobretodo en imágenes con un alto contenido de desorden, es decir que contienen muchos objetos y elementos además del o los objetos de interés, por ejemplo, todo aquello que se presente en una calle transitada o la diversa cantidad de escombros en una zona de emergencia post-desastre.

El ***filtro Gaussiano*** suaviza los detalles de una imagen, obteniendo una representación de diversas regiones homogéneas de los objetos presentes, es decir que suaviza texturas en función de la intensidad de los pixeles contenidos en el operador $f\,(m,n)$ de (3.1). El *núcleo* del filtro se define como

$$w_{gauss}\,(m,n) = g\,(m,n,\sigma) = \frac{1}{2\pi\sigma^2}\cdot\exp\left[-\frac{m^2+n^2}{2\sigma^2}\right] \tag{3.3}$$

donde σ es la *desviación estándar*, considerando $\sigma_m = \sigma_n$ como el factor que regula el efecto del filtro junto con el tamaño del núcleo. Al incrementar estos factores, los detalles que definen la periferia de los objetos tienden a perderse, siempre y cuando σ adquiera un valor que atenué los valores extremos del núcleo, por ejemplo $\sigma = 1$ define un apreciable efecto de suavizado para un núcleo de 5×5.

Si consideramos que la distribución Gaussiana bidimensional tiene medias $\mu_m,\ \mu_n \neq 0$ y desviaciones estándar diferentes $\sigma_m \neq \sigma_n$, (3.3) deja de ser uniforme y es expresado de forma completa como

$$w_{gauss_{ext}}\,(m,n,\sigma_m,\sigma_n) = g_{ext}\,(m,n) = \frac{1}{2\pi\sigma_m\sigma_n}\cdot\exp\left[\frac{-(m-\mu_m)^2}{2\sigma_m^2}+\frac{-(n-\mu_n)^2}{2\sigma_n^2}\right] \tag{3.4}$$

El efecto de suavizado se obtiene porque en el dominio de la frecuencia, (3.3) se comporta como un filtro paso bajas, atenuando también ruidos en la imagen que superen su frecuencia de corte. Este operador es utilizado normalmente como paso previo en algunos algoritmos extractores de características, como describirán en las secciones posteriores de este Capítulo.

3.2.2 Filtros detectores de bordes

Una de las fuentes comunes para formar descriptores de personas, son los ***mapas de bordes*** definidos por el contraste de la imagen, en los límites de los objetos, que los límites de los objetos por la diferencia de intensidades entre los pixeles de diversas regiones, utilizando la operación (3.1) a una imagen con filtros *paso altas* en el dominio de la frecuencia.

El concepto de ***bordes***, se fundamento en la teoría de cálculo diferencial al especificarse como relaciones de cambio. Inicialmente la diferenciación se planteó entre un par de pixeles paralelos, en dirección horizontal y vertical como

$$\frac{\partial \mathrm{I}}{\partial i} = \mathrm{I'}_i\,(i,j) = \mathrm{I}\,(i,j) - \mathrm{I}\,(i,j+1) \qquad \forall i \in 1,H; \quad j \in 1,W-1 \tag{3.5a}$$

$$\frac{\partial \mathrm{I}}{\partial j} = \mathrm{I'}_j\,(i,j) = \mathrm{I}\,(i,j) - \mathrm{I}\,(i+1,j) \qquad \forall i \in 1,H-1; \quad j \in 1,W \tag{3.5b}$$

en donde $\mathrm{I'}_i\,(i,j)$ es un elemento del *mapa de bordes* horizontales y $\mathrm{I'}_j\,(i,j)$ es uno de los verticales. Las diferenciaciones (3.5a) y (3.5b) se comprobaron por *series de Taylor* y de dicho análisis se encontró que al operar con dos elementos adyacentes, se obtenía una mejor aproximación de la derivada parcial [66], redefiniendo (3.5) de la siguiente manera

$$\mathrm{I'}_i\,(i,j) = \mathrm{I}\,(i,j-1) - \mathrm{I}\,(i,j+1) \tag{3.6a}$$

$$\mathrm{I'}_j\,(i,j) = \mathrm{I}\,(i-1,j) - \mathrm{I}\,(i+1,j) \tag{3.6b}$$

expresiones que al ser interpretadas como vectores, representan un par de plantillas que señalan la posición de operandos como

$$w_i = \begin{bmatrix} 1 & 0 & -1 \end{bmatrix}^T \tag{3.6c}$$

$$w_j = w_i^T \tag{3.6d}$$

donde la magnitud del borde del pixel (i,j) corresponde al elemento central del vector.

Prewitt en su artículo *Realce y extracción de objetos* [67], analizó las aproximaciones de las derivadas en imágenes (3.6), con el operador ***gradiente***. Como parte de sus resultados, obtuvo una mejor estimación de las diferenciaciones mediante mínimos cuadrados al operar regiones de pixeles de un tamaño de 3×3, utilizando (3.1) con los siguientes núcleos denominados ***operadores de Prewitt***

$$w_{prewitt_i} = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \tag{3.7a}$$

$$w_{prewitt_j} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \tag{3.7b}$$

Afirmó que al considerar más pixeles adyacentes, se evita obtener bordes falsos por la presencia de ruido en la imagen, en comparación con las plantillas (3.6c) y (3.6d). La primera descripción de los operadores (3.7), se realizó en 1966 en [68], sin embargo la referencia común es del artículo [67] de 1970.

En 1968, Irwin Sobel y Gary Feldman desarrollaron otro operador para calcular el *gradiente*, considerando la *distancia Manhattan* [65] entre los pixeles de cada vector ortogonal; dos diagonales en común y la horizontal o vertical que define cada mapa de bordes [69], en la región f de (3.1). De esta formulación determinaron los núcleos de pesos conocidos como ***operadores Sobel-Feldman***

$$w_{sobel_i} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad (3.8a) \qquad w_{sobel_j} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \qquad (3.8b)$$

Una extensión del planteamiento de los operadores (3.8) incluye suavizar los bordes de forma Gaussiana, al calcular los coeficientes de los núcleos de dimensiones superiores a 3×3. Esta metodología independiente del planteamiento original, se describe en [66].

Los operadores de Prewitt (3.7) y Sobel (3.8), han sido la base de otros trabajos enfocados en obtener un operador óptimo para calcular el *gradiente*, como en la disertación post-doctoral de Scharr Hanno [70], en donde plantea una metodología para obtener núcleos de tamaño impar a partir de 3×3 elementos, denominados como ***operadores Scharr*** cuyo par base es el siguiente

$$w_{scharr_i} = \begin{bmatrix} -3 & 0 & 3 \\ -10 & 0 & 10 \\ -3 & 0 & 3 \end{bmatrix} \qquad (3.9a) \qquad w_{scharr_j} = \begin{bmatrix} 3 & 10 & 3 \\ 0 & 0 & 0 \\ -3 & -10 & -3 \end{bmatrix} \qquad (3.9b)$$

Estos núcleos demostraron estimar las correspondientes derivadas direccionales con una mejor precisión, comparados con los operadores (3.8).

Para obtener el ***gradiente de la imagen***, se calculan las derivadas parciales aplicando la convolución (3.1) con algún par de los operadores (3.7), (3.8) ó (3.9). Con esta información, se obtiene el gradiente de cada elemento formado por su magnitud $M\,(i,j)$ y orientación espacial $\theta\,(i,j)$ como

$$M\,(i,j) = \sqrt{\mathrm{I'}_i\,(i,j)^2 + \mathrm{I'}_j\,(i,j)^2} \qquad (3.10a)$$

$$\theta\,(i,j) = \arctan\left(\mathrm{I'}_i\,(i,j)/\mathrm{I'}_j\,(i,j)\right) \qquad (3.10b)$$

Ambos datos forman el ***mapa de gradientes o bordes orientados*** $\nabla \mathrm{I}\,(i,j)$, los cuáles son utilizados en ciertos descriptores que se exponen en la Sección 3.3.

Existen otros planteamientos mas específicos para la detección de bordes de una imagen, como el algoritmo de Canny que es mas selectivo por la combinacion de algunos filtros mencionados anteriormente.

La metodología consiste de los siguientes pasos; primero suaviza la imagen con un filtro (3.3), después calcular las derivadas parciales, y proceder a seleccionar solo aquellos bordes que si pertenezcan a un contorno por medio de la técnica de *no máxima supresión*, mayores a cierta magnitud M y que su orientación los conecte con otros en el mapa [71].

3.2.3 Filtros del espacio escala

La variación de tamaño del o los objetos a detectar, ocasionada por su naturaleza como el caso de la estatura de las personas o por la distancia entre los objetivos y la cámara cuando se captura la escena, trasciende en la formación de un modelo de referencia.

Una de las maneras de representar un *imagen en diferentes escalas*, es reproducir el contenido inicial disminuyendo la cantidad de detalles estructurales y de texturas, aparentando obtener proyecciones a diferentes alejamientos. Este efecto es conseguido al suavizar y conservar las fronteras de los objetos en el cuadro, al aplicar cambios de resolución de la imagen original en combinación con alguno de los filtros paso banda que se presentan a continuación.

El operador ***Laplaciano del Gaussiano*** abreviado LoG en inglés [72], calcula las segundas derivadas parciales del núcleo (3.3), obteniendo un filtro que combina ambos bordes definido de la siguiente forma

$$\begin{aligned} w_{LoG}(m,n) = \nabla^2 g(m,n,\sigma) &= \frac{\partial^2 g(m,n,\sigma)}{\partial k^2} + \frac{\partial^2 g(m,n,\sigma)}{\partial l^2} \\ &= \frac{1}{2\pi\sigma^2} \cdot \left[\frac{m^2+n^2-2\sigma^2}{\sigma^4}\right] \cdot exp\left[-\frac{m^2+n^2}{2\sigma^2}\right] \\ &= \left[\frac{m^2+n^2-2\sigma^2}{\sigma^4}\right] \cdot g(m,n,\sigma) \end{aligned} \tag{3.11}$$

Al filtrar una imagen con el operador (3.11), se obtiene una representación a escala formada por regiones homogéneas denominados *blobs*, cuyos valores son atenuados en la banda del filtro definida por un círculo centrado en el origen de radio $\sqrt{2\sigma}$, que determina las diferenciaciones de segundo orden en los cruces por cero. Para obtener el efecto completo del núcleo LoG, se recomienda que el tamaño de los operadores $n \times n$ en (3.1), se defina por el entero impar más cercano de la relación $n \geq 6\sigma$.

Debido al número de operaciones involucradas para calcular LoG, Marr y Hildreth en [72], demostraron que (3.11) puede aproximarse mediante una ***diferencia de Gaussianas***, abreviada DoG en inglés con la siguiente expresión

$$\begin{aligned} w_{DoG}(m,n,\sigma,\rho) = \mathcal{D}(m,n,\sigma) &= g(m,n,\rho\sigma) - g(m,n,\sigma) \\ &= \frac{1}{2\pi\sigma(\rho\sigma-\sigma)} \cdot \left[\exp\left(-\frac{m^2+n^2}{2\rho\sigma^2}\right) - \exp\left(-\frac{m^2+n^2}{2\sigma^2}\right)\right] \end{aligned} \tag{3.12}$$

donde ρ es una constante proporcional que relaciona la diferencia de suavizado a partir de σ. El operador (3.12) es una *aproximación del gradiente de segundo orden* de la imagen suavizada, consecuentemente al aplicarlo en una imagen por la convolución, se obtiene un **mapa de bordes**, definido en dos escalas determinadas por ρ.

Los conjuntos de estos mapas de bordes, submuestreados en resolución forman las denominadas **pirámides de octavas**, que definen un conjunto de *características en el espacio escala*, razón que llevó a trascender el operador DoG más que el LoG, además del costo de calculo involucrado.

3.3 Descriptores de características

Son algoritmos que especifican conjuntos de *puntos característicos*, detectados en imágenes representativas del o los objetos de interés, con la finalidad de construir *patrones* que permitan reconocer los objetivos, mediante una metodología de *clasificación*.

Esta Sección reseña los ***descriptores*** más utilizados para la detección de personas, implementados en circunstancias semejantes a las consideradas en este proyecto, algunos de estos trabajos son; el seguimiento de peatones en ambientes altamente transitados, reconocimiento de objetos modelados por partes o en sistemas auxiliares de actividades de búsqueda y rescate.

3.3.1 Histogramas de gradientes orientados, HOG

Método que representa la estructura espacial del contenido de una proyección 2D, es decir la forma o figura de los objetos a cuadro en una imagen. Aunque no son los autores del concepto, Dalal y Triggs popularizaron el uso de *Histogramas de Gradientes Orientados*, HOG en inglés, al utilizarlos en su trabajo dedicado a la detección de peatones [18].

El descriptor consiste en una colección de *histogramas normalizados*, que representan las orientaciones que predominan en la descripción de la silueta del objeto, representadas en un vector característico de 3780 componentes. A continuación se expone una síntesis del procedimiento en el Algoritmo 3.3.1, considerando como base la explicación del descriptor HOG original [18], además de las presentadas en [40, 66].

Algoritmo 3.3.1: Descriptor de Histogramas de Gradientes Orientados, HOG. –*Parte I*–

Entrada(s): $f\,(m,n)$ ⇒ imagen o fragmento de 128×64 pixeles
Salida(s): v ⇒ vector característico extraído de $f\,(m,n)$ con 1×3780 componentes

▷ Cálculo del gradiente usando (3.1) y operadores (3.7)
▷ La magnitud se calcula con (3.10a) y la orientación con (3.10b) en el rango de 0 a 180°
$f'_i = f\,(p,q) * w_{prewitt_i}\,(p,q)\,;\qquad f'_j = f\,(p,q) * w_{prewitt_j}\,(p,q)$
$M\,(m,n)\,,\theta\,(m,n) \leftarrow \nabla f\,(m,n)$
▷ Cálculo de histogramas de orientaciones por bloques de 8×8 pixeles
para $r = 1$ *a* $r = 16$ **hacer** ; con $\Delta r = 1$ por iteración
    $lren = [8r-7, 8r] \rightarrow [lren_1, lren_2]$
    para $s = 1$ *a* $s = 8$ **hacer** ; con $\Delta s = 1$ por iteración
        $lcol = [8s-7, 8s] \rightarrow [lcol_1, lcol_2]$
        $M_{loc}\,(r,s) \leftarrow M\,(lren_1 : lren_2, lcol_1 : lcol_2)$
        $\theta_{loc}\,(r,s) \leftarrow \theta\,(lren_1 : lren_2, lcol_1 : lcol_2)$
        $H\,\{r,s\} \leftarrow hist\_orient\,(M_{loc}, \theta_{loc}, bins = 9)$
    fin
fin

Algoritmo 3.3.1: Descriptor de Histogramas de Gradientes Orientados, HOG. –*Parte II*–

```
15 ▷ Normalización de histogramas por vecindarios de 2 × 2
16 para p = 1 a p = 15 hacer ; con Δp = 1 por iteración
17 |   para q = 1 a q = 7 hacer ; con Δq = 1 por iteración
18 |   |   H_n{p,q} ← normHog(H{p,q})
19 |   fin
20 fin
21 ▷ Vector característico final
22 v ← concatenar_histogramas(H_n)
```

Considerando como referencia la numeración de renglones del Algoritmo 3.3.1, a continuación se presentan observaciones y comentarios adicionales de los métodos representados como funciones de un programa, que forman parte de la metodología HOG.

1 La dimensión de $f(m,n)$, puede variar de acuerdo a la aplicación por ejemplo, para la detección de peatones se trabaja la relación 2:1 o su inverso 1:2.

2 En la línea 4, el gradiente también puede ser calculado por los operadores (3.8) ó (3.9). Para imágenes con más de un canal, como las representadas en el espacio de color RGB, la magnitud del gradiente corresponde a la componente mayor de los canales.

3 El histograma local del renglón 12, se forma por nueve bloques o *bins* que dividen el rango de las orientaciones. Cada gradiente contribuye con su magnitud al bloque que le corresponde acorde a su orientación.

4 Dalal y Triggs concluyeron por experimentación que la normalización más funcional, se obtiene al considerar 2×2 o 3×3 regiones de histogramas de $H\{p,q\}$ [18]. El método de la línea 18, aplica la *norma Euclidiana*, también conocida como **L2**, al vector formado al concatenar los histogramas $H\{p+1,q\}$, $H\{p,q+1\}$, $H\{p+1,q+1\}$ y el seleccionado en cada iteración de los ciclos.

5 Finalmente, el vector característico v es resultado de concatenar los 105 vectores de 36 componentes cada uno, obtenidos en la normalización.

El procedimiento 3.3.1 es aplicado a cada imagen para obtener un conjunto de vectores característicos, que posteriormente son analizados por un clasificador, como en el detector de personas reportado en [18], en donde emplean una *Máquina de Soporte Vectorial*, SVM para comparar el modelo obtenido del entrenamiento.

Al analizar $f(m,n)$ por regiones y vecindarios, se proporciona cierta tolerancia al vector característico, para detectar variaciones estructurales. Otra particularidad del descriptor HOG es su independencia a las variaciones de contraste, porque no utiliza los valores de intensidades de los pixeles de forma directa.

3.3.2 Transformada de características invariantes a escala, SIFT

Describe el contenido de una imagen con un conjunto de puntos distintivos, cada uno definido por un vector de 132 componentes que corresponden; a su posición determinada por coordenadas cartesianas, la escala donde se detectó, un ángulo tendencia de orientación y 128 elementos que describen el vecindario que rodea al punto característico.

Esta metodología fue propuesta por Lowe en [19] y esta organizada en cuatro etapas principales; detección de puntos extremos en el *espacio escala*, localización de puntos clave, asignación de orientación y la formación del vector descriptivo local. El desarrollo de cada una de estas fases se sintetiza en el Algoritmo 3.3.2, a partir de la publicación original y las reseñas [40, 42, 66, 73].

Algoritmo 3.3.2: Transformada de características invariantes a escala, SIFT. –*Parte I*–

Entrada(s): $f\,(i,j) \Rightarrow$ imagen a describir de resolución $W \times H$
$\sigma_{in} = 0.8$; $\rho = 2$; $\delta_{in} = 0.5$; $D_{ref} = 3 \times 10^{-2}$; $r = 10$; $\lambda_{ori} = 1.5$; $\lambda_d = 8$
$n\_octavas = 4$; $n\_niveles = 3$

Salida(s): Puntos característicos de f; $Pc_f = \{pc_1, pc_2, \dots, pc_k\} \mid pc_k = [i_k, j_k, \sigma_k, \theta_k, vd_k]$

▷ Detección de puntos extremos
$f_p\,(i,j) \leftarrow aumentar\_resolución\,(f, 2)$
para $o = 1$ ***hasta*** $o = n\_octavas$ **hacer** ; con $\Delta o = 1$ por iteración
  para $s = 1$ ***hasta*** $s = n\_niveles$ **hacer** ; con $\Delta s = 1$ por iteración
    $\sigma_o = o \cdot \sigma_{in}$
    $piram\_dog\,\{o,s\} \leftarrow f_p\,(m,n) * \left[g\left(m,n,\rho^{1/s} \cdot \sigma_o\right) - g\left(m,n,\rho^{1/(s-1)} \cdot \sigma_o\right)\right]$
  fin
  $f_a\,(m_a, n_a) = reducir\_resolución\left(f_p, 2\right)$; $\Rightarrow$ $ajustar\_resolución\left(f_p, f_a\right)$
fin
$pts\_ext\left\{x_q\right\} \leftarrow obtener\_max\_min\,(piram\_dog,\, \delta_{in})$; donde $x_q = \left[m_q, n_q, \sigma_q\right]$
▷ Localización de puntos característicos
para cada x_q ***de*** pts_ext **hacer**
  $\hat{x_q'} = -\frac{\partial^2 D(x_q')}{\partial {x_q'}^2}^{-1} \frac{\partial D(x_q')}{\partial x_q'} = -H_{x_q'}{}^{-1} \frac{\partial D(x_q')}{\partial x_q'}$; donde $\hat{x_q'} = \left[\hat{m_q}, \hat{n_q}, \hat{\sigma_q}\right]$ es el reajuste de x_q
  si $\left(\left|\hat{m_q}\right| \Vert \left|\hat{n_q}\right| \Vert \left|\hat{\sigma_q}\right|\right) < 0.5$ **entonces**
    $D\left(\hat{x_q'}\right) = D\left(x_q\right) + \frac{1}{2} \frac{\partial D}{\partial x_q'}^T \hat{x_q'}$
    si $\left|D\left(\hat{x_q'}\right)\right| > D_{ref}$ **entonces**
      $H_p = \begin{bmatrix} D_{mm} & D_{mn} \\ D_{mn} & D_{nn} \end{bmatrix}$; $Tr\left(H_p\right) = D_{mm} + D_{nn}$; $Det\left(H_p\right) = D_{mm}D_{nn} - {D_{mn}}^2$
      si $\frac{Tr(H_p)^2}{Det(H_p)} < \frac{(r+1)^2}{r}$ **entonces**
        $xa_q = x_q + \hat{x_q'} = [m_a, n_a, \sigma_a]^T$; $\Rightarrow$ $i_k, j_k = interpolar\_coordenadas\left(xa_q\right)$
        $\sigma_k = xa_q\,[\sigma_a]$; $\Rightarrow$ $Pc_f\,\{pc_k\} = [i_k, j_k, \sigma_k]$; $\triangleright pc_k = [i_k, j_k, \sigma_k]$
fin

Algoritmo 3.3.2: Transformada de características invariantes a escala, SIFT. –*Parte II*–

▷ Asignación de orientación(es) dominante(s)
para cada pc_k ***de*** Pc_f **hacer**
  $l_v = \lceil 3\lambda_{ori}\sigma_k \rfloor$; $x = [[i_k - l_v, i_k + l_v]] = [x_1, x_2]$; $y = [[j_k - l_v, j_k + l_v]] = [y_1, y_2]$
  si $0 \leq x, y \leq W, H$ **entonces**
    $L(i,j) = f(i,j) * g(p,q,\sigma_k)$; ⇒ $vg(r,c) = L(x_1 : x_2, y_1 : y_2)$
    $M(r,c), \theta_{vg}(r,c) \leftarrow \nabla vg(r,c)$; ⇒ $M_g(r,c) = M(r,c)\, g(r,c,\lambda_{ori}\sigma_k)$
    $H_{pc_k} \leftarrow hist\_orient\left(M_g, \theta_{vg}, bins = 36\right)$; ⇒ $\theta_k(l) \leftarrow max\_orient\_bins\left(H_{pc_k}\right)$
    para cada l ***de*** θ_k **hacer** $Pc_f\{pc_k\} \leftarrow adjuntar(\theta_k(l))$ ▷ $pc_k = [i_k, j_k, \sigma_k, \theta_k]$
fin

▷ Descriptor local de pc_k, vector vd_k de 1×128
para cada pc_k ***de*** Pc_f **hacer**
  $L(i,j) = f(i,j) * g(m,n,\sigma_k)$
  ▷ Cálculo del vecindario de gradientes de 16×16 elementos
  para $p = -\lambda_d$ ***hasta*** $p = \lambda_d - 1$ **hacer** ; con $\Delta p = 1$
    para $q = -\lambda_d$ ***hasta*** $q = \lambda_d - 1$ **hacer** ; con $\Delta q = 1$
      $a = [p + 0.5, q + 0.5]$; $b = [i_k, j_k]$; $v^T = {\sigma_k}^{-1}\mathbf{R}_{\theta_k} a^T + b = \left[\left[i_p, j_p\right]\right]^T = \left[v_i, v_j\right]^T$
      $vg(p,q) = \left[L\left(v_i, v_j - 1\right) - L\left(v_i, v_j + 1\right), L\left(v_i - 1, v_j\right) - L\left(v_i + 1, v_j\right)\right] = \left[L'_{i_p}, L'_{j_p}\right]$
    fin
  fin
  $M_{vg}(p,q) = g(p,q,\lambda_d\sigma_k)\, \|vg(p,q)\|$; $\theta_{vg}(p,q) = \arctan\left(vg(p,q)\right) - \theta_k$
  para $r = 1$ ***hasta*** $r = 4$ **hacer** ; con $\Delta r = 1$ por iteración
    $lr = [4(r-1) + 1, 4r] \rightarrow [lr_1, lr_2]$
    para $c = 1$ ***hasta*** $c = 4$ **hacer** ; con $\Delta c = 1$ por iteración
      $lc = [4(c-1) + 1, 4c] \rightarrow [lc_1, lc_2]$
      $M_h(r,c) \leftarrow M_{vg}(lr_1 : lr_2, lc_1 : lc_2)$; $\theta_h(r,c) \leftarrow \theta_{vg}(lr_1 : lr_2, lc_1 : lc_2)$
      $H_{pc_k}(r,c) \leftarrow hist\_orient\left(M_h, \theta_h, bins = 8\right)$
    fin
  fin
fin
$vd_k \leftarrow concatenar\_histogramas\left(H_{pc_k}\right)$; ⇒ $Pc_f\{pc_k\} \leftarrow adjuntar(vd_k)$
▷ $pc_k = [i_k, j_k, \sigma_k, \theta_k, vd_k]$

Con base en la numeración de renglones del Algoritmo 3.3.2, en seguida se explica el procedimiento expuesto en ecuaciones y funciones que forma la metodología SIFT.

• Detección de puntos extremos

En esta etapa se buscan estos elementos utilizando una *pirámide de octavas*, formada por representaciones escala de $f(i,j)$ disminuyendo su tamaño en cada octava, comenzando con el doble de su resolución f_p. La pirámide se forma aplicando filtros de interpolación como el binomial o cúbico [42].

Las octavas se dividen en s niveles que corresponden a una f_p filtrada con (3.12), cambiando los coeficientes escala en cada imagen con base en σ_o, como se especifica en la línea 6.

El número de niveles y octavas especificado, es propuesto como mínimo para buscar los candidatos en la pirámide, particularmente en los niveles s con representaciones adyacentes $s \pm 1$. en estas imágenes se examinan los puntos centrales de regiones de 3×3 elementos, que se define al desplazar el vecindario por la imagen s con un paso horizontal y/o vertical determinado por octava

$$\delta_o = \delta_{in} 2^{o-1} \tag{3.13}$$

Un *punto extremo* es aquel centro del vecindario cuya magnitud sea menor o mayor que todos sus vecinos, incluyendo las regiones del mismo tamaño ubicadas en la misma posición en los niveles $s \pm 1$. La función declarada en la línea 10, obtiene la lista de todos los puntos extremos encontrados, cada uno determinado por su posición (m_q, n_q) relativa a f_p, y por el factor escala

$$\sigma_q = \frac{\delta_o}{\delta_{in}} \sigma_{in} 2^{s/n\_niveles} \tag{3.14}$$

• Localización de puntos característicos

Después de la lista de pts_ext se seleccionan aquellos que cumplan dos condiciones. El primer requisito se fundamenta en un *análisis de curvatura* [39], desarrollando la expansión del operador (3.12), mediante la *serie de Taylor* hasta el término cuadrático considerando como origen la posición de x_q, de tal modo que la serie define el espacio $x'_q = x - x_q = [m, n, \sigma]$ tal que

$$\mathcal{D}\left(x_q\right) = \sum_{\eta=0}^{2} \frac{D^{\eta}\left(x'_q\right)}{\eta!} {x'_q}^{\eta} = D\left(x_q\right) + \frac{\partial D^T}{\partial x'_q} x'_q + \frac{1}{2} {x'_q}^T \frac{\partial^2 D}{\partial {x'_q}^2} x'_q \tag{3.15}$$

al derivar (3.15) e igualar el resultado a cero, se obtiene un refinamiento de la posición y escala del punto x_q, denotado como $\hat{x'_q}$ calculado en la línea 13, en donde la segunda derivada parcial es manejada como la ***matriz Hessiana*** de $D\left(x'_q\right)$, definida como

$$H_{x'_q} = \begin{bmatrix} D_{mm} & D_{mn} & D_{m\sigma} \\ D_{mn} & D_{nn} & D_{n\sigma} \\ D_{m\sigma} & D_{n\sigma} & D_{\sigma\sigma} \end{bmatrix} \tag{3.16}$$

en donde las diferenciaciones de la matriz (3.16) se obtienen como las aproximaciones (3.6), pero considerando un espacio de tres dimensiones de tal modo que algunas de las derivadas parciales son

$$D_{mm} = \frac{\partial^2 D}{\partial m^2} = D\left(m+1, n, \sigma\right) + D\left(m-1, n, \sigma\right) - 2D\left(m, n, \sigma\right) \tag{3.17a}$$

$$D_{nn} = \frac{\partial^2 D}{\partial n^2} = D\left(m, n+1, \sigma\right) + D\left(m, n-1, \sigma\right) - 2D\left(m, n, \sigma\right) \tag{3.17b}$$

$$D_{\sigma\sigma} = \frac{\partial^2 D}{\partial \sigma^2} = D\left(m, n, \sigma+1\right) + D\left(m, n, \sigma-1\right) - 2D\left(m, n, \sigma\right) \tag{3.17c}$$

$$D_{mn} = \frac{\partial^2 D}{\partial m \partial n} = \frac{1}{4}\begin{bmatrix} D\,(m+1, n+1, \sigma) + D\,(m+1, n-1, \sigma) \\ +D\,(m-1, n+1, \sigma) + D\,(m-1, n-1, \sigma) \end{bmatrix} \quad (3.17d)$$

$$D_{m\sigma} = \frac{\partial^2 D}{\partial m \partial \sigma} = \frac{1}{4}\begin{bmatrix} D\,(m+1, n, \sigma+1) + D\,(m+1, n, \sigma-1) \\ +D\,(m-1, n, \sigma+1) + D\,(m-1, n, \sigma-1) \end{bmatrix} \quad (3.17e)$$

$$D_{n\sigma} = \frac{\partial^2 D}{\partial n \partial \sigma} = \frac{1}{4}\begin{bmatrix} D\,(m, n+1, \sigma+1) + D\,(m, n+1, \sigma-1) \\ +D\,(m, n-1, \sigma+1) + D\,(m, n-1, \sigma-1) \end{bmatrix} \quad (3.17f)$$

Las diferenciales restantes son semejantes al conjunto (3.17), resultando en un sistema de ecuaciones lineal que define $\hat{x_q'}$. Entonces, solo si todos los ajustes de x_q son menores de 0.5, el análisis continúa calculando $D(\hat{x_q'})$ con la expresión proviene de sustituir $\hat{x_q'}$ en (3.15), indicada de la línea 15.

$\mathcal{D}\left(\hat{x_q'}\right)$ debe ser mayor que la magnitud de referencia D_{ref}, reportada en [19] para descartar *puntos extremos* procedentes de ruido. Esta es la primer condición señalada en el renglón 16, al aprobarse se analizan las *principales curvaturas* [39] de $\mathcal{D}\left(\hat{x_q'}\right)$, en función de la proporción de magnitud r de los eigenvalores asociados a la posición del punto x_q, definida con respecto a la matriz H_p.

La segunda condición descarta los *puntos extremos* cuyos eigenvalores, tengan una relación mayor a la especificada en la línea 18, considerando $r = 10$ de acuerdo a [19]. Superando los dos filtros la terna de valores x_q se reajusta sumando $\hat{x_q'}$, para interpolar los datos al dominio de $f\,(i,j)$ con la función del renglón 19, obteniendo finalmente un *punto característico* pc_k.

• Asignación de orientación dominante

La tercera etapa del algoritmo SIFT, relaciona la dirección tendencia θ_k de los gradientes que rodean a cada pc_k. El vecindario cuadrado vg que define θ_k, se centra en (i_k, j_k) y sus dimensiones dependen de σ_k, por ello se revisa que los puntos límite de vg existan en el espacio $\mathbb{N}^{W\times H}$.

Al cumplirse la condición del renglón 24, se extrae $vg\,(r,c)\,\epsilon\mathbb{N}^{l_v\times l_v}$ redondeando l_v al entero impar más cercano, a partir de $f\,(i,j)$ filtrada por (3.3) con σ_k. Después se calcula ∇vg, ponderando su magnitud por una *distribución Gaussiana*, como se muestra en la línea 27.

La información de M_g y θ_{vg} se simplifica en un histograma de orientaciones H_{pc_k}, distribuyendo $\theta_{vg}\epsilon\,[0, 2\pi]$ en 36 partes. Para formarlo se busca la división de mayor acumulación h_{bp} y las otras dos más cercanas h_{bp-} y h_{bp+}, de manera que cada elemento del histograma se define como

$$\theta_k\,(l) = \frac{2\pi\left(h_{bp} - 1\right)}{36} + \frac{\pi}{36}\left(\frac{h_{bp-} - h_{bp+}}{h_{bp-} - 2h_{bp} + h_{bp+}}\right) \quad (3.18)$$

Sí existen otras divisiones de H_{pc_k} con un acumulado mayor a $0.8h_{bp}$, se crea otro punto característico definiendo su ángulo con el mismo procedimiento descrito en el párrafo anterior,

razón por la que el índice l señala cada dirección dominante encontrada al implementar la función de búsqueda de la línea 28.

Finalmente en el renglón 29 se adjunta cada $\theta_k\ (l)$ encontrada, a la tercia que define cada punto pc_k, reiterando que es posible crear otro punto en caso de encontrar más de un ángulo, con los mismos datos que pc_k pero distinguidos por cada orientación dominante.

• **Descriptor local del punto característico**

Este último elemento de pc_k es un vector formado al concatenar 16 histogramas de gradientes orientados, calculados de un vecindario de 16×16 elementos, centrado en la posición de cada punto característico y orientado en dirección θ_k. Cada punto v de esta región vg, se rota e interpola al espacio de f en la línea 37 con la matriz de giro

$$\mathbf{R}_{\theta_k} = \begin{bmatrix} \cos\theta_k & -\sin\theta_k \\ \sin\theta_k & \cos\theta_k \end{bmatrix} \tag{3.19}$$

Después con (3.6) se calculan las derivadas parciales de primer orden, considerando la imagen f suavizada por el filtro (3.3) con σ_k. Ambos valores del gradiente definen un punto del vecindario $vg\ (p, q)$, expresado como (3.10) pero ponderando su magnitud como se indica en el renglón 41.

Con base en M_{vg} y θ_{vg} se calculan los 16 histogramas de gradientes orientados H_{pc_k}, de ocho componentes cada uno. En el Algoritmo 3.3.2 se divide M_{vg} en 16 partes de 4×4 puntos y se extrae un histograma por cada sección, acumulando la magnitud de cada elemento en el intervalo donde se encuentre su ángulo, proceso que es especificado en las líneas 42 a 50.

El descriptor de histogramas del $vg\ (p, q)$ se plantea diferente en [19], pero no especifica su proceso, en contraste con [73] donde se expone otro método completamente descrito, sin embargo ambas formulaciones no son prácticas por sus operaciones involucradas.

Y finalmente los histogramas del arreglo $H_{pc_k}\ (r, c)$ se concatenan en un vector vd_k, que se adjunta al arreglo de datos que define cada punto característico pc_k, concluyendo la descripción de una región de la imagen f usando el planteamiento de SIFT.

3.3.3 Speed-up robust features, SURF

La capacidad descriptiva del algoritmo SIFT fue considerada como la mejor en su momento para tareas como; el reconocimiento de objetos, reconstrucción tridimensional, formación de imágenes panorámicas, entre otras que inspiraron a optimizar y mejorar el método, por la cantidad de operaciones que involucra, pretendiendo llevarlo a aplicaciones en tiempo real.

Uno de los métodos alternos más trascendente a SIFT fue propuesto por Bay en [74], denominado como ***Speed-Up Robust Features*** identificado por sus siglas SURF. Su planteamiento es muy similar a SIFT, pero este se formuló con la idea de reducir operaciones y simplificar algunas etapas, como se observa en la siguiente síntesis del procedimiento realizada con base en [74, 75].

Algoritmo 3.3.3: Speed-Up Robust Features, SURF. *–Parte I–*

Entrada(s): $f\,(i,j) \Rightarrow$ imagen a describir de resolución $W \times H$
$n\_octavas = 4; \quad n\_niveles = 4; \quad DoH_{ref} = 3 \times 10^2; \quad \alpha = 40$
Salida(s): Puntos característicos de f; $Pc_f = \{pc_1, pc_2, \dots, pc_k\} \mid pc_k = \left[i_k, j_k, L_k, \theta_k, vd_k, \mathcal{L}_{sign_k}\right]$

1 ▷ Determinación de puntos característicos
2 $F\,(i,j) = \sum_{1 \le i \le W} \sum_{1 \le j \le H} f\,(i,j)$ ▷ Integral de imagen
3 **para** $o = 1$ ***hasta*** $o = n\_octavas$ **hacer** ; con $\Delta o = 1$ por iteración
4 $\quad \Delta d = 2^{o-1}$
5 $\quad$ **para** $s = 1$ ***hasta*** $s = n\_niveles$ **hacer** ; con $\Delta s = 1$ por iteración
6 $\quad\quad L_k = 2^o s + 1$
7 $\quad\quad DoH_f\,\{o,s\} \leftarrow detector\_rápido\_hessiano\,(F, L_k, \Delta d)$ ▷ $p_k = \left[i_k, j_k, L_k, DoH^{L_k}\{f\,(i_k, j_k)\}\right]$
8 $\quad$ **fin**
9 **fin**
10 **para** $o = 1$ ***hasta*** $o = n\_octavas$ **hacer** ; con $\Delta o = 1$ por iteración
11 $\quad$ **para** $s = 2$ *y* 3 ***de*** DoH_f **hacer**
12 $\quad\quad$ **para cada** p_k ***de*** $DoH_f\,\{o,s\}$ **hacer**
13 $\quad\quad\quad$ **si** $p_k\left[DoH^{L_k}\{f\,(i_k, j_k)\}\right] > DoH_{ref}$ **entonces**
14 $\quad\quad\quad\quad$ **si** $maximo\_en\_vecindario\,(p_k, 3, s)$ **entonces**
15 $\quad\quad\quad\quad\quad x_k = [i_k, j_k, L_k]^T; \quad x'_k = x - x_k = [i_b, j_b, L_b]^T$
16 $\quad\quad\quad\quad\quad \hat{x'_k} = -\frac{\partial^2 f(x'_k)}{\partial {x'_k}^2}^{-1} \frac{\partial f(x'_k)}{\partial x'_k} = -{H_{x'_k}}^{-1} \frac{\partial f(x'_k)}{\partial x'_k}; \mid \hat{x'_k} = [i_b, j_b, L_b]^T$
17 $\quad\quad\quad\quad\quad$ **si** $(i_b \parallel j_b \parallel 0.5 L_b) < 2^{o-1}$ **entonces**
18 $\quad\quad\quad\quad\quad\quad x_k = x_k + \hat{x'_k} = [i_k + i_b, j_k + j_b, L_k + L_b]^T = [i_k, j_k, L_k]^T$
19 $\quad\quad\quad\quad\quad\quad Pc\,\{pc_k\} \leftarrow adjuntar\,(x_k)\,;$ ▷$pc_k = [i_k, j_k, L_k]$
20 $\quad\quad$ **fin**
21 $\quad$ **fin**
22 **fin**
23 ▷ Asignación de orientación dominante
24 **para cada** pc_k ***de*** Pc **hacer**
25 $\quad \sigma_{L_k} = 0.4 L_k$
26 $\quad$ **para** $n = -3$ ***hasta*** $n = 2$ **hacer**
27 $\quad\quad$ **para** $m = -3$ ***hasta*** $m = 2$ **hacer**
28 $\quad\quad\quad v = \left[\left[\sigma_{L_k}(m + 0.5) + i_k\right], \left[\sigma_{L_k}(n + 0.5) + j_k\right]\right]; \quad g_w = g\left(m, n, 2.5\sigma_{L_k}\right)$
29 $\quad\quad\quad f'_i\,(v) = D_i^{L_k} * f\,(v) \cdot g_w; \quad f'_j\,(v) = D_j^{L_k} * f\,(v) \cdot g_w; \quad f'\,(m,n) = \left[f'_i, f'_j\right]$
30 $\quad\quad\quad \theta_{f_i} = \arctan\left(f'_i\right); \quad \theta_{f_j} = \arctan\left(f'_j\right); \quad \theta_f\,(m,n) = \left[\theta_{f_i}, \theta_{f_j}\right]$
31 $\quad\quad$ **fin**
32 $\quad$ **fin**
33 $\quad$ **para** $p = 1$ ***hasta*** $p = \alpha$ **hacer**
34 $\quad\quad \theta_p = 2\pi p/\alpha; \quad a_{\theta_p} = \left[\sum f'\left(m, n : {f'}_i\right), \sum f'\left(m, n : {f'}_j\right)\right] \quad sii \quad \forall \theta_f; \theta_{f_i}, \theta_{f_j} \epsilon \left[\theta_p \pm \pi/6\right]$
35 $\quad\quad \Phi\,\{p\} = \left[\left\|a_{\theta_p}\right\|, \angle a_{\theta_p}\right]$
36 $\quad$ **fin**
37 $\quad \theta_k = argmax\left(\left\|\Phi\left[a_{\theta_p}\right]\right\|\right); \quad Pc_f\,\{pc_k\} \leftarrow adjuntar\,(\theta_k)$ ▷ $pc_k = [i_k, j_k, L_k, \theta_k]$
38 **fin**

Algoritmo 3.3.3: Speed-Up Robust Features, SURF. –*Parte II*–

▷ Descripción local de pc_k, vector vd_k de 1×64 componentes
para cada pc_k ***de*** Pc_f **hacer**
  ▷ Cálculo de gradientes en vecindario de $20\sigma_{L_k} \times 20\sigma_{L_k}$
  para $p = -10$ ***hasta*** $p = 9$ **hacer** ; con $\Delta p = 1$
    para $q = -10$ ***hasta*** $q = 9$ **hacer** ; con $\Delta q = 1$
      $a = [p + 0.5, q + 0.5]$; $b = [i_k, j_k]$; $\sigma_{L_k} = 0.4L_k$; $v^T = \sigma_{L_k}\mathbf{R}_{\theta_k}a^T + b$
      $v = \left[\left[i_p, j_p\right]\right]$; $g_m = g\left(p, q, 3.3\sigma_{L_k}\right)$
      $f_i'(v) = D_i^{L_k} * f(v) \cdot g_m$; $f_j'(v) = D_j^{L_k} * f(v) \cdot g_m$; $f' = \left[f_i'(v), f_j'(v)\right]^T$
      $\left[f_p', f_q'\right]^T = \mathbf{R}_{-\theta_k} f'$; $vg(p,q) = \left[f_p', f_q'\right]$
    fin
  fin
  para $r = 1$ ***hasta*** $r = 4$ **hacer**
    $y = [5r - 4, 5r] = [y_1, y_2]$
    para $c = 1$ ***hasta*** $c = 4$ **hacer**
      $x = [5c - 4, 5c] = [x_1, x_2]$
      $vr\{r,c\} = \sum_{m=x_1, n=y_1}^{x_2, y_2} \left[vg\left(m, n : f_p'\right), vg\left(m, n : f_q'\right), \left|vg\left(m, n : f_p'\right)\right|, \left|vg\left(m, n : f_q'\right)\right|\right]$
    fin
  fin
  $vd_k \leftarrow norm\_12\,(concatenar\,(vr))$
  $\beta = 1/{L_k}^2$; $\mathcal{L}\{f(i_k, j_k)\} = \beta\left[D_{ii}^{L_k} f(i_k, j_k) + D_{jj}^{L_k} f(i_k, j_k)\right]$
  si $(\mathcal{L}\{f(i_k, j_k)\} \geq 0) \Rightarrow \mathcal{L}_{sign_k} = 1$; de lo contrario $\mathcal{L}_{sign_k} = 0$
  $Pc_f\{pc_k\} \leftarrow adjuntar\left(vd_k, \mathcal{L}_{sign_k}\right)$ ▷ $pc_k = \left[i_k, j_k, L_k, \theta_k, vd_k, \mathcal{L}_{sign_k}\right]$
fin

Para mantener el análisis de imágenes en el espacio escala, utilizaron la ***integral de imagen*** y el estimador DoG (3.12) para reducir el tiempo de proceso.A continuación se describe cada etapa del método, con base en el desarrollo del Algoritmo 3.3.3.

• Determinación de puntos característicos

Considerando una imagen en escala de grises $f(i,j)$, el primer paso del método consiste en calcular la *integral de imagen* F, que es una aproximación de la operación convolución como se describirá más adelante. F es una matriz de puntos en donde cada uno es igual a la suma de todas las intensidades de los pixeles anteriores a la abscisa y ordenada del punto seleccionado para su cálculo en f [40].

Para SURF se elige el punto $f(i_W, j_H)$ para obtener la ***integral de imagen***, obteniendo una matriz del mismo tamaño que f donde la esquina superior izquierda $F(1,1)$ tiene el valor del pixel seleccionado para el cálculo. La esquina inferior derecha de $F(i_W, j_H)$, es igual a la suma de intensidades de todos los pixeles anteriores al punto $f(i_W, j_H)$ (operación del renglón 2).

Durante el desarrollo del algoritmo se forman vecindarios de tamaño variable, razón por la que es común aplicar técnicas de *padding* a la imagen de trabajo, una de las opciones es por extensión de simetría [66].

Los *puntos característicos* son buscados en el espacio escala y son seleccionados por tres criterios, que se examinan en la última de dos etapas de elección que se describen a continuación:

1 *Mapas de puntos en el espacio escala*

La reducción de operaciones de SURF se fundamenta en la integral de imagen y en las aproximaciones propuestas del operador LoG, formadas por estimaciones de las derivadas parciales Gaussianas $\partial^2 g/\partial i^2$, $\partial^2 g/\partial j^2$ y $\partial g/\partial i \partial j$ considerando como base un filtro (3.3) de 9×9 elementos y $\sigma_0 = 1.2$.

Estas diferenciales Gaussianas Bay cita definirlas en [74] con la transformada discreta Wavelet Haar, sin embargo no describe el método. Ante este dilema, Oyallon en [75] determina las derivadas usando conjuntos de tamaño y escala variable en función de $L_k \epsilon \mathbb{N}$, que los relaciona con la *pirámide de octavas* del *espacio escala*, de tal manera que las bases de aproximación al operador LoG son las siguientes

$$\frac{\partial^2 g}{\partial i^2} \cong D_{ii}^{L_k}(m,n) = \begin{cases} -2; & \text{si } (m,n)\,\epsilon \Lambda_1 \cap \Lambda_2 \\ 1; & \text{si } (m,n)\,\epsilon \Lambda_1 \setminus \Lambda_2 \\ 0; & \forall \text{ caso diferente} \end{cases} \quad donde \quad \begin{matrix} \Lambda_1 = [-A \dots A, -B \dots B] \\ \Lambda_2 = [-C \dots C, -B \dots B] \end{matrix} \tag{3.20a}$$

$$\frac{\partial^2 g}{\partial j^2} \cong D_{jj}^{L_k}(m,n) = \begin{cases} -2; & \text{si } (m,n)\,\epsilon \Lambda_3 \cap \Lambda_4 \\ 1; & \text{si } (m,n)\,\epsilon \Lambda_3 \setminus \Lambda_4 \\ 0; & \forall \text{ caso diferente} \end{cases} \quad donde \quad \begin{matrix} \Lambda_3 = [-B \dots B, -A \dots A] \\ \Lambda_4 = [-B \dots B, -C \dots C] \end{matrix} \tag{3.20b}$$

$$\frac{\partial^2 g}{\partial i \partial j} \cong D_{ij}^{L_k}(m,n) = \begin{cases} -1; & \text{si } (m,n)\,\epsilon \Lambda_{NO} \cup \Lambda_{SE} \\ 1; & \text{si } (m,n)\,\epsilon \Lambda_{NE} \cup \Lambda_{SO} \\ 0; & \forall \text{ caso diferente} \end{cases} \quad donde \quad \begin{matrix} \Lambda_{NO} = [-L \dots 1, 1 \dots L] \\ \Lambda_{SE} = [1 \dots L, -L \dots -1] \\ \Lambda_{NE} = [1 \dots L, 1 \dots L] \\ \Lambda_{SO} = [-L \dots -1, -L \dots -1] \end{matrix} \tag{3.20c}$$

con

$$L_k = 2^o s + 1; \qquad A = \frac{3L_k - 1}{2}; \qquad B = L_k - 1; \qquad C = \frac{L_k - 1}{2}$$

donde o es la enésima octava de la pirámide y s el enésimo nivel de cada octava. Las operaciones entre los conjuntos $\Lambda_1, \dots, \Lambda_4$, Λ_{NO}, Λ_{SE}, Λ_{SO} y Λ_{NE} definen cada derivada como un operador en forma de *núcleo de convolución*, tomando como referencia el centro del vecindario.

A diferencia de SIFT, el descriptor SURF busca puntos candidatos a ser característicos incrementando el tamaño de los operadores (3.20) en cada nivel y octava de la pirámide, sin modificar las dimensiones de la imagen f.

Cada nivel s representa una escala diferente, pero en lugar de obtener toda la imagen suavizada, solo se examinan puntos muestreados cada $(i + \Delta d, j + \Delta d)\,\epsilon f$ posiciones, cambiando Δd en cada

octava como está indicado en la línea 4. El análisis de cada punto se hace con el **detector rápido Hessiano**, definido por el determinante de la *matriz Hessiana*

$$DoH^{L_k}\{f(i,j)\} = \frac{1}{{L_k}^4}\left[D_{ii}^{L_k} * f(i,j) \cdot D_{jj}^{L_k} * f(i,j) - \left\{0.912 \cdot D_{ij}^{L_k} * f(i,j)\right\}^2\right] \tag{3.21}$$

en el cual las operaciones convolución, son realizadas con la **integral de imagen** F simplificando el proceso a sumas y accesos a memoria con las siguientes expresiones

$$D_{ii}^{L_k} * f(i,j) = \left[U_{\Lambda_1} - 3U_{\Lambda_2}\right] * f(i,j) \tag{3.22a}$$

$$D_{jj}^{L_k} * f(i,j) = \left[U_{\Lambda_3} - 3U_{\Lambda_4}\right] * f(i,j) \tag{3.22b}$$

$$D_{ij}^{L_k} * f(i,j) = \left[U_{\Lambda_{NE}} + U_{\Lambda_{SO}} - U_{\Lambda_{NO}} - U_{\Lambda_{SE}}\right] * f(i,j) \tag{3.22c}$$

donde la variable U representa un vecindario de distribución uniforme, delimitado por la región subíndice y definido en forma general por la siguiente expresión

$$U_\Lambda(m,n) = \begin{cases} 1; & \text{si } (m,n)\,\epsilon\Lambda \\ 0; & \forall \text{ caso diferente} \end{cases} \quad donde \quad \Lambda = [\lambda_1 \dots \lambda_2, \lambda_3 \dots \lambda_4] \tag{3.23}$$

de tal forma que las convoluciones entre un punto de la imagen $f(i,j)$ y las diferentes U_Λ, son expresadas en términos de $F(i,j)$ considerando (3.23), definidas en términos generales como

$$\begin{aligned} U_\Lambda * f(i,j) = & F(i-\lambda_1, j-\lambda_3) + F(i-\lambda_2-1, j-\lambda_4-1) \\ & - F(i-\lambda_1, j-\lambda_4-1) - F(i-\lambda_2-1, j-\lambda_3) \end{aligned} \tag{3.24}$$

por consecuencia, cada una de las expresiones (3.22) se definen de forma finita al desarrollarlas considerando (3.24), determinando el cálculo del **detector rápido Hessiano** en términos de la integral de imagen, proceso indicado en la línea 7 donde $DoH_f\{o,s\}$ contiene un conjunto de puntos por cada nivel de la pirámide de octavas, definiendo cada punto como un vector $p_k = \left[i_k, j_k, L_k, DoH^{L_k}\{f(i_k, j_k)\}\right]$.

2 *Selección de puntos característicos*

Esta etapa consiste en revisar que cada punto p_k cumpla con *tres condiciones* para ser característico. El *primer requisito* indicado en la línea 13 es que el determinante de la matriz Hessiana (3.21) del punto examinado, tenga un magnitud mayor a una referencia como la declarada en el Algoritmo 3.3.3, definida por experimentación en [74].

La *segunda condición* señalada en la línea 14, verifica que $\left\|p_k\left[DoH^{L_k}\{f(i_k,j_k)\}\right]\right\|$ sea el máximo en un vecindario cúbico de $3\times3\times3$ pixeles, como en el método aplicado para encontrar máximos y mínimos en el Algoritmo 3.3.2, pero en este caso solo considerando máximos.

Y el *tercer requerimiento* en la línea 17 es la revisión de magnitud del ajuste de la posición y escala del punto p_k, calculado por el análisis de curvatura de la misma forma que en SIFT, considerando la variable L_k, las derivadas parciales (3.20) y las de primer orden que se definen más adelante.

Al cumplirse las condiciones, se ajusta la tercia inicial x_k al sumarle $\hat{x_k'}$, renglón 18, definiendo parcialmente cada punto característico pc_k del conjunto descriptor Pc, declarado en la línea 19.

• Asignación de orientación dominante

La tercer parte del algoritmo, determina un ángulo θ_k, al que tiende el vecindario de gradientes que rodea al punto pc_k. La vinculación con la orientación dominante, hace invariante a rotaciones la detección de los puntos, incrementando la capacidad descriptiva.

El vecindario v analizado para obtener θ_k es de $6\sigma_{L_k} \times 6\sigma_{L_k}$ elementos, centrado en (i_k, j_k), donde σ_{L_k} es el factor escala de la función base (3.3) que define las representaciones de la pirámide de octavas, por esta razón se define como función de L_k como se declara en la línea 25.

La derivada parcial de primer orden de cada elemento del vecindario v, también se calcula aplicando la integral de imagen como el caso de (3.22), mediante las siguientes expresiones para cada componente dimensional

$$D_i^{L_k} * f\,(i_k, j_k) = \left[U_{\Lambda_5} - U_{\Lambda_6}\right] * f\,(i_k, j_k) \quad \text{donde} \quad \begin{matrix} \Lambda_5 = [-l_k \dots -1, -l_k \dots l_k] \\ \Lambda_6 = [1 \dots l_k, -l_k \dots l_k] \end{matrix} \tag{3.25a}$$

$$D_j^{L_k} * f\,(i_k, j_k) = \left[U_{\Lambda_7} - U_{\Lambda_8}\right] * f\,(i_k, j_k) \quad \text{donde} \quad \begin{matrix} \Lambda_7 = [-l_k \dots l_k, -l_k \dots -1] \\ \Lambda_8 = [-l_k \dots l_k, 1 \dots l_k] \end{matrix} \tag{3.25b}$$

en donde $l_k = \lceil 0.8 L_k \rceil$. Considerando (3.23) y (3.24), al desarrollar las ecuaciones (3.25) se escriben en términos de la integral de imagen con respecto al punto (i_k, j_k) y sus vecinos de esta forma

$$D_i^{L_k} * f\,(i_k, j_k) = \begin{matrix} F\,(i_k + l_k, j_k + l_k) + F\,(i_k, j_k - l_k - 1) - F\,(i_k + l_k, j_k - l_k - 1) \\ - F\,(i_k, j_k + l_k) - F\,(i_k - 1, j_k + l_k) \\ - F\,(i_k - l_k - 1, j_k - l_k - 1) + F\,(i_k - 1, j_k - l_k - 1) \\ + F\,(i_k - l_k - 1, j_k + l_k) \end{matrix} \tag{3.26a}$$

$$D_j^{L_k} * f\,(i_k, j_k) = \begin{matrix} F\,(i_k + l_k, j_k + l_k) + F\,(i_k - l_k - 1, j_k) - F\,(i_k + l_k, j_k) \\ - F\,(i_k - l_k - 1, j_k + l) - F\,(i_k + l_k, j_k - 1) \\ - F\,(i_k - l_k - 1, j_k - l_k - 1) + F\,(i_k + l_k, j_k - l_k - 1) \\ + F\,(i_k - l_k - 1, j_k - 1) \end{matrix} \tag{3.26b}$$

La posición de cada elemento del vecindario v definida en la línea 28, se utiliza para calcular (3.26a) y (3.26b), ponderando los resultados con el correspondiente punto de una distribución (3.3) del mismo tamaño de v y con $\sigma = 2.5\sigma_{L_k}$, especificado en la línea 29. De cada derivada se obtiene un ángulo con la función arco tangente, conservando las dos magnitudes y orientaciones en las matrices $f'\,(m, n)$ y $\theta_f\,(m, n)$.

Para definir la orientación dominante, primero se obtiene una representación vectorial $\Phi\{p\}$ de magnitud-dirección en función de α secciones del rango $[0, 2\pi]$. En especifico, por cada división definida como θ_p, se forma un vector a_{θ_p} cuyos elementos son la suma por componente dimensional, de los gradientes del vecindario cuyo ángulo θ_{f_i} o θ_{f_j}, esté en el intervalo de $\theta_p \pm \pi/6$, como se indica en la línea 34.

Finalmente para asignar la orientación dominante θ_k, los vectores del arreglo a_{θ_p} se transforman a su representación polar indicada como $\Phi\{p\}$. Entonces, θ_k es el ángulo relacionando al vector de mayor magnitud encontrado en a_{θ_p}. Dicho proceso esta descrito en la línea 37, en conjunto con la indexación de θ_k a pc_k.

• **Descripción local del punto característico**

Similar al Algoritmo 3.3.2, cada punto característico también es relacionado con un vector local descriptivo, vd_k pero en SURF es de 64 componentes. La definición de vd_k comienza en un vecindario de gradientes vg, de $20\sigma_{L_k}\times 20\sigma_{L_k}$ con centro en $pc_k\ (i_k, j_k)$ y orientado a la dirección θ_k.

En el renglón 44 se define un punto v de vg por iteración, el cual se rota multiplicándolo por (3.19) y se interpola al espacio de f. Después, en la línea 46, se calculan las derivadas parciales aplicando (3.26) con respecto al punto v transformado, ponderando las diferenciales con g_m y colocando los resultados en el vector f', para regresar los gradientes al espacio de vg y asignarlos a su respectivo punto del vecindario, como se señala en el renglón 47.

Al disponer completamente de vg, se procede a seccionar el vecindario de gradientes en bloques de 5×5 puntos, para extraer un vector vr por cada división, teniendo un total de 16 vectores con cuatro componentes cada uno, definidas por sumas de derivadas parciales, como se especifica en la línea 54.

Finalmente, todos los elementos del arreglo $vr\{r, c\}$ se concatenan en un solo vector y se normaliza, para definir vd_k de cada pc_k. Adicionalmente, para optimizar la coincidencia de puntos característicos, se calcula el *Laplaciano* $\mathcal{L}\{f\ (i_k, j_k)\}$ con las expresiones del renglón 58, para conservar su signo $\mathcal{L}_{sign_k}$ al adjuntarlo junto con vd_k al arreglo que define cada pc_k, obteniendo su definición completa mostrada en la línea 60.

3.3.4 Red neuronal convolucional, CNN

Algoritmo de *aprendizaje automático* diseñado para *reconocer patrones* en estructuras de datos, que tienen dependencias espaciales en regiones locales de la malla. La metodología transforma la información de entrada a un dominio de mayor dimensión, en donde se definen representaciones singulares correspondientes a los diferentes objetivos que se desean identificar [76].

Continuando con el enfoque de este Capítulo, en esta Sección se presentan los conceptos involucrados en la *extracción de características* de imágenes, que puede hacer una CNN. Para esto, primero se expone una síntesis de los conceptos base implicados, relacionados con las redes neuronales abreviadas NN en inglés.

Modelo de neurona artificial

Es un planteamiento matemático que representa el procesamiento de señales llevado a cabo por una neurona biológica, que es el principal elemento de una NN. A pesar de que existen diferentes esquemas que describen este proceso, el modelo de neurona ***McCulloch-Pitts*** propuesto en

1943, es el más utilizado en las aplicaciones actuales de *aprendizaje profundo* [77].

La primera fase del modelo, cuantifica la ***sinapsis total*** o *estimulo de activación* z de la neurona k, equivalente a una *combinación lineal* de la señal de entrada x, que puede ser de índole escalar, vectorial, matricial o *tensorial*, la cual es ponderada por coeficientes denominados como ***pesos sinápticos*** w [78], de la siguiente manera

$$z_k = \sum_{j=1}^{m} w_j x_j + b_k; \qquad j \epsilon \mathbb{N} \mid j = 1, 2, \dots, m \tag{3.27a}$$

en donde el subíndice j, señala de forma general los elementos que componen un estímulo x_d y cada uno de los pesos w_k de la neurona, ambas variables para efectos prácticos de la explicación serán considerados como vectores de m elementos, tal que $x_d = [x_1, x_2, \dots, x_m]^T$ y $w_k = [w_1, w_2, \dots, w_m]^T$. Y b_k es un ajuste escalar adicional de la activación.

La segunda etapa de la *neurona*, acota o transforma (3.27a) a otro rango de valores, de acuerdo a la aplicación de la neurona k, utilizando una ***función de activación o transferencia*** que obtiene la respuesta final y_k, definida por la siguiente expresión

$$y_k = \Upsilon\left(w_k{}^T x_d + b_k\right) = \Upsilon\left(z_k\right) \tag{3.27b}$$

La neurona puede ***predecir*** o ***clasificar*** señales, dependiendo del tipo de función $\Upsilon(\cdot)$ que se ajuste a la tarea deseada. Algunas de las relaciones matemáticas que se aplican a z_k, son las siguientes [76]:

1 ***Heaviside*** define solo dos tipos de respuesta (3.27b), considerando un intervalo de magnitud que define el criterio de *clasificación* de la salida, por ejemplo

$$\Upsilon_H\left(z_k\right) = \begin{cases} 1 & si \quad z_k \geq 0 \\ -1 & si \quad z_k < 0 \end{cases} \tag{3.28a}$$

2 ***Identidad*** se usa para predecir un número real y conserva la linealidad de (3.27a).

$$\Upsilon_{\mathrm{I}}\left(z_k\right) = \mathrm{I}\left(z_k\right) = z_k \tag{3.28b}$$

3 ***Sigmoidea*** acota la salida en el intervalo $y_k \epsilon [0, 1]$, razón por la que es utilizada para representar *porcentajes de certidumbre* o *probabilidades* de una respuesta deseada, utilizando la siguiente expresión

$$\Upsilon_\sigma\left(z_k\right) = \frac{1}{1 + e^{-z_k}} \tag{3.28c}$$

4 ***Unidad de Rectificación Lineal, ReLu*** determina la parte positiva de (3.27a), fue propuesta como una aproximación de (3.28c) para facilitar la determinación de w_k, para una aplicación afín, y se define como

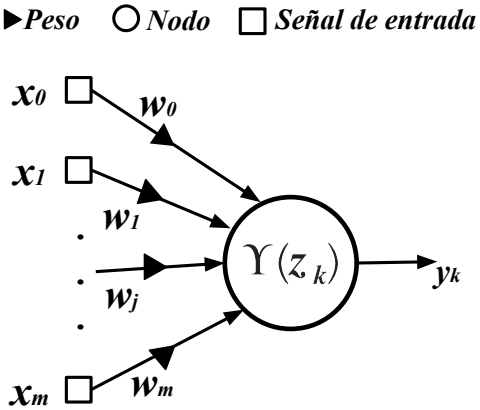
$$\Upsilon_R\left(z_k\right) = max\left(0, z_k\right) \tag{3.28d}$$

Los *pesos sinápticos* w_k de la neurona, actúan como coeficientes de *filtros* del estímulo de entrada x_d, por esta razón se definen de acuerdo al índole de los datos a procesar. Al trabajar con imágenes, se utilizan conjuntos de neuronas interconectadas que funcionan como ***bancos de filtros***, que interactúan como se describe a continuación.

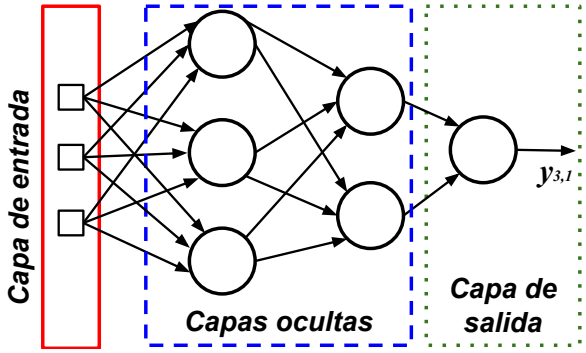
Redes neuronales

Son configuraciones de conexión entre conjuntos de *neuronas*, de manera que las respuestas de salida de un grupo son las señales de entrada de otras, determinando una estructura jerárquica de enlaces. Considerando la definición (3.27), las *redes neuronales* se pueden interpretar como una *composición de funciones simples* (3.27b), que forman una más compleja y_k [76].

Las conexiones en una red y el flujo de las señales a través de ella, se describen convencionalmente en ***grafos***, en donde el elemento base es la representación simbólica del modelo (3.27), que se muestra en la Figura 3.1a. En este emblema el coeficiente de ajuste es incluido en w_k, como $w_0 = b_k$, si y solo si también se extiende x_d con $x_0 = 1$, permitiendo expresar (3.27a) como una operación ***convolución*** [78].



(a) Representación simbólica de una neurona.



(b) División de capas de una red neuronal; grafo de NN totalmente conectada fc.

Figura 3.1. Esquema de una neurona y de la distribución de conexiones en una red.

En los *grafos* la representación de los *pesos sinápticos*, normalmente se omite por la cantidad de conexiones entre los nodos de la red, resaltando la estructura de los enlaces denominada como ***arquitectura***, que es clasificada de acuerdo al flujo de las señales en el *grafo* en dos tipos [78]:

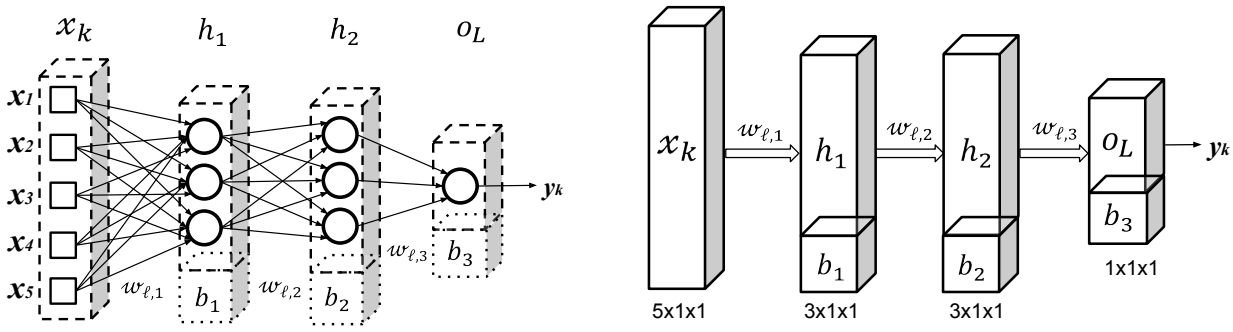
- ***Feedforward***; identificada por tener una sola dirección de propagación, como se observa en la Figura 3.1b.
- ***Recurrentes***; incorporan a las redes *feedforward* por lo menos una conexión de retroalimentación.

Sin importar la *arquitectura*, las neuronas de una red son organizadas en L *capas*, identificadas de acuerdo a su ubicación en el *grafo*, como se muestra en la Figura 3.1b. Cada *capa* puede tener un diferente número de neuronas o mantener constante dicho factor, de acuerdo a la aplicación.

Al manejar redes con ***capas ocultas***, los parámetros w_k y b_k convencionalmente son manejados y expresados en arreglos tipo matricial, de manera que cada columna representa el conjunto de pesos y coeficientes de ajuste de una capa ℓ, y la correspondiente intersección en cada renglón, corresponde al grupo variables de una neurona de la capa, es decir

$$\mathcal{W} = \begin{bmatrix} w_{1,1} & \dots & w_{\ell,1} & \dots & w_{L,1} \\ \vdots & & \vdots & & w_{L,2} \\ \vdots & & \vdots & & \vdots \\ w_{1,n_\ell-1} & & \vdots & & \vdots \\ w_{1,n_\ell} & \dots & w_{\ell,d} & \dots & w_{L,n_\ell} \end{bmatrix} \quad (3.29) \qquad \mathcal{B} = \begin{bmatrix} b_{1,1} & \dots & b_{\ell,1} & \dots & b_{L,1} \\ \vdots & & \vdots & & w_{L,2} \\ \vdots & & \vdots & & \vdots \\ b_{1,n_\ell-1} & & \vdots & & \vdots \\ b_{1,n_\ell} & \dots & b_{\ell,d} & \dots & b_{L,n_\ell} \end{bmatrix} \quad (3.30)$$

en donde n_ℓ indica la n-enésima neurona de cada capa L que forman la *arquitectura*. Esta notación concede la flexibilidad para declarar escalares, vectores, matrices o ***tensores*** de información a procesar en la NN, y adicionalmente permite representar las *capas* como *bloques*, que simplifican los diagramas de estructuras extensas, como se ejemplifica en los siguientes esquemas básicos



(a) Conversión del grafo de una NN a bloques. (b) Diagrama en bloques de una NN.

Figura 3.2. Representación de una red neuronal *feedforward* como diagrama de bloques.

Las *capas* en las Figuras 3.2 son representadas como prismas rectangulares para denotar la capacidad de manejar arreglos de datos de n dimensión. Los coeficientes de ajuste b_k son indicados como otro poliedro que se incorpora a las respectivas *capas*, etiquetadas para distinguir los *bloques*, que son marcados con el tamaño del arreglo que procesa.

De acuerdo al desarrollo de nuevas *arquitecturas*, se han adecuado los diagramas de la Figura 3.2 de diversas formas, para representar nuevas propuestas, operadores adicionales y sobretodo conservar su legibilidad a dichas modificaciones.

El diseño de la red trasciende en su funcionamiento, porque los parámetros $\mathcal{W}$ y $\mathcal{B}$ de las *capas ocultas*, definen un ***espacio de abstracción*** al que transforman la señal de entrada (como un *descriptor* de características), para favorecer su análisis y obtener la respuesta de salida y_d que puede ser un escalar o vector, de acuerdo el propósito de la NN [76].

Después de especificar una *arquitectura*, el siguiente paso es encontrar una combinación de valores (3.29) y (3.30), aplicando la metodología base que se describe a continuación.

Aprendizaje de la red

Es el procedimiento que determina un conjunto $\mathcal{W}$ y $\mathcal{B}$, para predecir o clasificar información especifica usando una NN. Este proceso también es conocido como ***entrenamiento*** y consiste en estimar los parámetros mencionados, mediante ajustes de acuerdo a un grupo de señales ejemplares, aplicando principalmente alguno de los siguientes modos de aprendizaje [79]:

- ▷ ***Supervisado***; técnica que utiliza señales de entrada asociadas con sus respectivas respuestas, que se esperan obtener en la salida de la red neuronal.
- ▷ ***No supervisado***; este método ajusta los parámetros $\mathcal{W}$ y $\mathcal{B}$, al procesar muestras representativas que ingresan a la red, con el objetivo de encontrar patrones en los datos.

El uso de una *red neuronal* como ***descriptor*** de características de imágenes, se fundamenta en el *espacio de representación* integrado por los *pesos sinápticos* y *coeficientes de ajuste*, que se determinan por medio de un *entrenamiento supervisado*.

A partir de una colección de imágenes $\Lambda = \{\mathrm{I}_j\} \mid j\epsilon\mathbb{N}; j = 1, \dots, N_\Lambda$, organizadas de acuerdo a los α objetivos a reconocer por la red neuronal. El *aprendizaje inspeccionado* utiliza un $\mathrm{T} \subset \Lambda$ denominado como conjunto de *entrenamiento*, cuyos elementos consisten de N_T observaciones asociadas con la respuesta esperada por la NN, es decir

$$\mathrm{T} = \left\{\left(\mathrm{I}_j, \mathcal{y}_j\right)\right\}_{j=1}^{N_\mathrm{T}} \tag{3.31}$$

Con base en la interpretación matemática de una red neuronal, el *aprendizaje vigilado* trata de determinar una función cuya respuesta $\mathcal{y}'$ sea la mejor aproximación al resultado deseado $\mathcal{y}$, definido por la última capa de la *arquitectura*, que tiene por señales de entrada son producto del proceso realizado por la composición de funciones de las capas antecesoras, que integran el *descriptor*.

En particular para los objetivos del proyecto la CNN es utilizada como un *clasificador* de imágenes, en donde cada objeto a reconocer es interpretado como una distribución de características, que se definen por los $\mathcal{W}$ y $\mathcal{B}$ de la red. Al propagarse una señal de entrada x_d, se obtiene una proporción de certidumbre en cada uno de los conjuntos de referencia.

Por consiguiente, la respuesta $\mathcal{y}'_d$ es interpretada como el conjunto de probabilidades o evaluaciones de semejanza, de x_d con las diferentes α clases. Este resultado es comparado con el esperado $\mathcal{y}_d$ para cada observación, mediante la función de ***entropía cruzada*** [80], determinada para un estímulo d como

$$\mathbb{L}_d = -\sum_{c=1}^{\alpha} \mathcal{y}_{d,c} \log \mathcal{y}'_{d,c} \tag{3.32a}$$

en donde el subíndice c señala la clase que se está comparando con la respuesta deseada. Si se restringe el modelo a dos categorías de clasificación, (3.32a) se reescribe, desarrollando para

$\alpha = 2$ de la siguiente manera

$$\mathbb{L}_d\left(y_d, y'_d\right) = -\left(\underbrace{y_d \log y'_d}_{c=1} + \underbrace{(1-y_d)\log\left(1-y'_d\right)}_{c=2}\right) \tag{3.32b}$$

esta expresión es denominada como ***entropía cruzada binaria***, considerando como base del logaritmo el número e . Entonces para el conjunto de T, se define la ***función de verosimilitud, costo o error*** total, utilizada como medida de ajuste para realizar el *entrenamiento*, y es determinada como

$$J\left(\mathcal{W}', \mathcal{B}'\right) = \frac{1}{N_\mathrm{T}} \sum_{d=1}^{N_\mathrm{T}} \mathbb{L}_d\left(y_d, y'_d\right) \tag{3.33}$$

La desemejanza (3.32), es producto del desacierto que aporta cada neurona en la red, debido a sus coeficientes ${w'}_{L,n_\ell}$ y ${b'}_{L,n_\ell}$. Para minimizar este error, primero se cuantifica la contribución que hace cada una de las neuronas de la capa, mediante el algoritmo de ***retro-propagación*** o ***backpropagation***[1] [76].

Este método trata de encontrar la mejor combinación de (3.29) y (3.30), a partir de un conjunto de valores iniciales $\mathcal{W}'_0$ y $\mathcal{B}'_0$ que son rectificados mediante un *proceso iterativo* de acuerdo al error de cada neurona y sus estímulos. El ajuste de los parámetros pretende minimizar la *función de costo* (3.33), mediante la técnica de ***gradiente descendiente*** que modifica los *pesos sinápticos* y coeficientes de ajuste en cada repetición, con la siguiente expresión

$$\mathcal{W}'_i = \mathcal{W}'_{i-1} - \gamma \nabla J_{\mathcal{W}}\left(\mathcal{W}'_{i-1}, \mathcal{B}'_{i-1}\right) \qquad i \epsilon \mathbb{N}; \quad i = 1, 2, \ldots, \mathcal{I} \tag{3.34}$$

donde el subíndice i señala la iteración de cálculo, $\nabla J_{\mathcal{W}}$ es la matriz de derivadas parciales del conjunto de pesos previo $\mathcal{W}'_{i-1}$, del cual cada elemento es usado para actualizar los nuevos pesos $\mathcal{W}'_i$, considerando un coeficiente ponderador γ denominado ***tasa de aprendizaje*** [77]. Para los coeficientes de ajuste se sustituye en (3.34), $\mathcal{B}'$ por $\mathcal{W}'$.

La funcionalidad de una *red neuronal* depende en gran medida de su *arquitectura* y *entrenamiento*, este último proceso puede necesitar de una gran cantidad de pares T, ocasionando el incremento de operaciones y memoria necesarios para calcular la *función de costo* (3.33) y ajustar los pesos sinápticos (3.34) y coeficientes de ajuste.

Sin embargo, al hacer dichos cálculos no se garantiza que el error disminuya a un valor deseado, por esta razón se determinan un límite de iteraciones $\mathcal{I}$. La causa de esta eventualidad, radica en los denominados ***hiper parámetros*** de la red [76], algunos de ellos son los valores iniciales[2] de $\mathcal{W}'_0$ y $\mathcal{B}'_0$, al cantidad de capas y de neuronas en cada una, el valor de γ en (3.34), las N_T observaciones, entre otros.

El primer desarrollo del algoritmo *backpropagation*, reajustaba $\mathcal{W}'$ y $\mathcal{B}'$ al evaluar (3.33) para cada T_j, ocasionando que los resultados de la red, no fueran aceptables. Por ello surgieron

[1] El método es ampliamente descrito en las Secciones 1.3 y 3.2 en [76]
[2] En el Capítulo 3, páginas 125-127 de [28] se describe una técnica para iniciar los pesos y coeficientes de ajuste.

propuestas, en donde el ajuste se realiza después de un ciclo de evaluaciones de muestras T_j, es denominado como ***epoch***. Los nuevos planteamientos hacen η *epochs* reordenando y/o dividiendo en subconjuntos a T, principalmente de las siguientes maneras [77]:

- ▷ *Disponiendo estocásticamente de las muestras*; que consiste en cambiar el orden de los elementos de T de forma aleatoria, para que en cada *epoch*, el ajuste (3.34) tienda a converger al mínimo. Esta variante es conocida como en inglés como ***Stochastic Gradient Descent***, SGD.

- ▷ *Usando grupos de observaciones*; esta estrategia fragmenta T en ρ subconjuntos, con el mismo número de elementos, para actualizar $\mathcal{W}'$ y $\mathcal{B}'$ al terminar de evaluar cada grupo ρ. Esta táctica es denominada como ***Mini-batch Gradient Descent***.

A su vez estas técnicas ocupan ajustes más específicos que (3.34), en las cuales se realizan cambios adaptativos de la *tasa de aprendizaje* considerando registros del comportamiento del error y de reajustes realizados. Estos métodos son llamados como ***optimizadores*** [76] y algunos de ellos son:

• ***Decaimiento progresivo***; disminuye γ en relación al número de ajustes aplicados y un decremento gradual de la tasa de aprendizaje. Las dos formas más comunes son el *decaimiento exponencial* (3.35a) y el *de tiempo inverso* (3.35b)

$$\gamma_\eta = \gamma_{\eta-1} e^{-\upsilon\eta} \qquad (3.35a) \qquad\qquad \gamma_\eta = \frac{\gamma_{\eta-1}}{1+\eta\upsilon} \qquad (3.35b)$$

donde la variable υ regula el decrecimiento de la *tasa de aprendizaje* y η denota el ***epoch***, dejando el término i para cada iteración del cálculo de (3.33) por cada muestra x_d [28].

• ***RMSprop***; reajusta los valores de $\mathcal{W}'$ y $\mathcal{B}'$ de acuerdo a la estabilidad que presenta la dirección del gradiente del cálculo previo $\eta - 1$. De la observación de dicho comportamiento, se busca que las magnitudes de las derivadas parciales tiendan a ser minúsculas, porque esto indica que la orientación del gradiente se esta encauzando a un punto mínimo, determinando en principio valores óptimos de $\mathcal{W}'$ y $\mathcal{B}'$, calculados en el proceso iterativo por las siguientes expresiones:

$$\mathcal{W}'_\eta = \mathcal{W}'_{\eta-1} - \gamma \frac{\nabla J_{\mathcal{W}}\left(\mathcal{W}'_{\eta-1}, \mathcal{B}'_{\eta-1}\right)}{\sqrt{S_{\mathcal{W}'_\eta} + \varepsilon}} \qquad (3.36a)$$

$$\mathcal{B}'_\eta = \mathcal{W}'_{\eta-1} - \gamma \frac{\partial J_{\mathcal{B}}\left(\mathcal{W}'_{\eta-1}, \mathcal{B}'_{\eta-1}\right)}{\partial b} \frac{1}{\sqrt{S_{\mathcal{B}'_\eta} + \varepsilon}} \qquad (3.36b)$$

$$S_{\mathcal{W}'_\eta} = \beta S_{\mathcal{W}'_{\eta-1}} + (1-\beta)\, \nabla J_{\mathcal{W}}\left(\mathcal{W}'_{\eta-1}, \mathcal{B}'_{\eta-1}\right) \circ \nabla J_{\mathcal{W}}\left(\mathcal{W}'_{\eta-1}, \mathcal{B}'_{\eta-1}\right) \qquad (3.36c)$$

$$S_{\mathcal{B}'_\eta} = \beta S_{\mathcal{B}'_{\eta-1}} + (1-\beta)\, \frac{\partial J_{\mathcal{B}}\left(W'_{\eta-1}, \mathcal{B}'_{\eta-1}\right)}{\partial b} \circ \frac{\partial J_{\mathcal{B}}\left(W'_{\eta-1}, \mathcal{B}'_{\eta-1}\right)}{\partial b} \qquad (3.36d)$$

en donde ε es un escalar para evitar una indeterminación de la división, un valor convencional es $\varepsilon = 10e^{-8}$, el operador $\circ$ indica un producto elemento por elemento, β es el ponderador de las contribuciones de las derivadas parciales previas[3] (3.36c) y (3.36d), de modo que al ser de gran magnitud su aportación es menor para orientar el descenso del gradiente en los reajustes (3.36a) y (3.36b) [28].

• ***Estimación Adaptativa del Momento, Adam***; es una combinación de los optimizadores *rmsprop* y *momento del gradiente*, porque utiliza un promedio de las derivadas parciales previas, ponderado exponencialmente, logrando una regulación de las aportaciones del gradiente de los *peso sinápticos* y coeficientes de ajuste sin modificar γ, de tal manera que el reajuste de parámetros lo hace de la siguiente forma:

$$v^c_{\mathcal{W}',\eta} = \frac{v_{\mathcal{W}',\eta-1}}{1-\beta_1^{\eta}}; \qquad \Rightarrow \quad v_{\mathcal{W}',\eta-1} = \beta_1 v_{\mathcal{W}',\eta-1} + (1-\beta_1)\,\nabla J_{\mathcal{W}}\left(\mathcal{W}'_{\eta-1}, \mathcal{B}'_{\eta-1}\right) \tag{3.37a}$$

$$v^c_{\mathcal{B}',\eta} = \frac{v_{\mathcal{B}',\eta-1}}{1-\beta_1^{\eta}}; \qquad \Rightarrow \quad v_{\mathcal{B}',\eta-1} = \beta_1 v_{\mathcal{B}',\eta-1} + (1-\beta_1)\,\frac{\partial J_{\mathcal{W}}\left(\mathcal{W}'_{\eta-1}, \mathcal{B}'_{\eta-1}\right)}{\partial b} \tag{3.37b}$$

$$S^c_{\mathcal{W}',\eta} = \frac{S_{\mathcal{W}',\eta-1}}{1-\beta_2^{\eta}}; \qquad \Rightarrow \quad S_{\mathcal{W}',\eta-1} = (3.36c)|_{\beta=\beta_2} \tag{3.37c}$$

$$S^c_{\mathcal{B}',\eta} = \frac{S_{\mathcal{B}',\eta-1}}{1-\beta_2^{\eta}}; \qquad \Rightarrow \quad S_{\mathcal{B}',\eta-1} = (3.36d)|_{\beta=\beta_2} \tag{3.37d}$$

$$\mathcal{W}'_{\eta} = \mathcal{W}'_{\eta-1} - \gamma\frac{v^c_{\mathcal{W}',\eta}}{\sqrt{S^c_{\mathcal{W}',\eta}+\varepsilon}} \tag{3.37e}$$

$$\mathcal{B}'_{\eta} = \mathcal{W}'_{\eta-1} - \gamma\frac{v^c_{\mathcal{B}',\eta}}{\sqrt{S^c_{\mathcal{B}',\eta}+\varepsilon}} \tag{3.37f}$$

en donde el par (3.37a) y (3.37b) corresponden a los promedios del *momento del gradiente*, ponderados exponencialmente por β_1, en forma descendiente para reducir la *función de costo* (3.33). La aportación de *rmsprop* cuantificada por (3.37c) y (3.37d) actúan como factores de normalización, determinados también por la media exponencial ponderada en este caso por β_2. Y ε es una constante para evitar que el denominador de (3.37e) y (3.37f) pueda ser indeterminado [28].

De acuerdo a la técnica de *optimización* que se utilice para el proceso de *aprendizaje* de la red, los coeficientes de ponderación y/o regulación del método, también son considerados como *hiper parámetros* del modelo. En la Figura 3.3, se sintetiza el proceso de entrenamiento descrito previamente.

Al finalizar el entrenamiento, se pretende que la red pueda ser capaz de llevar a cabo su tarea sin embargo, regularmente esto no sucede en el primer experimento, porque la cantidad de combinaciones de *hiper parámetros* que se pueden usar. Por esta razón al finalizar cada epoch, se evalúa el desempeño de las estimaciones de $\mathcal{W}'$ y $\mathcal{B}'$, con otro subconjunto de Λ denominado como *conjunto de validación*

$$\mathcal{V} \subset \Lambda \mid \mathrm{T} \cap \mathcal{V} = \emptyset \quad \Rightarrow \mathcal{V} = \left\{\left(\mathrm{I}_j, \mathcal{Y}_j\right)\right\}_{j=1}^{N_{\mathcal{V}}} \tag{3.38}$$

[3]Esta forma de mesurar las aportaciones, es una variación de la técnica conocida como ***momento del gradiente*** que se describe en [76, 77]

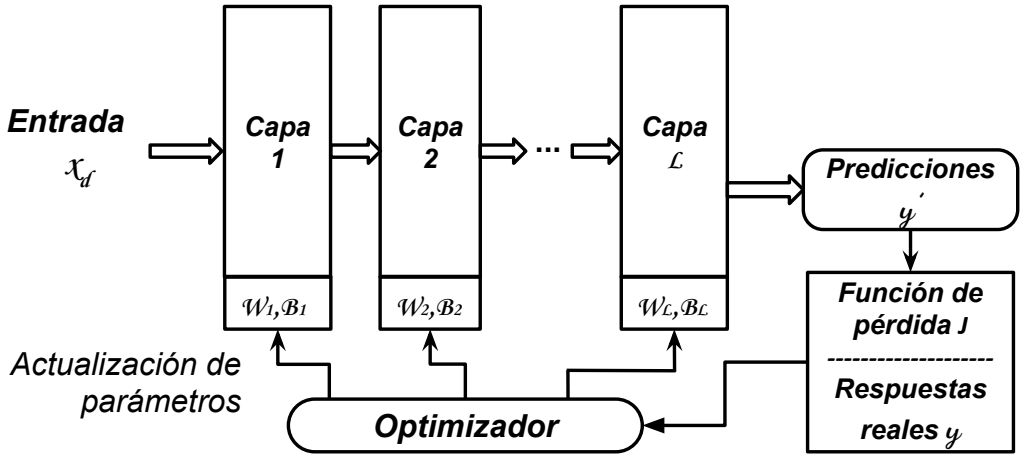


Figura 3.3. Diagrama de flujo del proceso de entrenamiento de una red *feedforward* [81].

De manera que, la colección de imágenes Λ o el conjunto de datos a procesar por la NN, normalmente se divide en tres subconjuntos en proporciones de 80/10/10 % del total de muestras N_Λ [80]. Dichos grupos son el de *entrenamiento* (3.31), *validación* (3.38) y *evaluación*, este último definido de la siguiente manera

$$\mathrm{E} \subset \Lambda \mid \mathrm{T} \cap \mathcal{V} \cap \mathrm{E} = \emptyset \quad \Rightarrow \mathrm{E} = \left\{\left(\mathrm{I}_j, y_j\right)\right\}_{j=1}^{N_\mathrm{E}} \tag{3.39}$$

El propósito de fragmentar el conjunto de datos, principalmente es para realizar variaciones de los *hiper parámetros* con la finalidad de mejorar las estimaciones de $\mathcal{W}$ y $\mathcal{B}$, después de cada ensayo de *entrenamiento* mesurando el desempeño de la NN, con el subconjunto (3.38).

La cantidad de reajustes durante el proceso de *aprendizaje* no esta limitado sin embargo, debe procurarse que la brecha entre el los errores de estimación entre T y $\mathcal{V}$ no sea mínima, porque los *pesos sinápticos* y *coeficientes de ajuste* tienden a definirse de forma especifica para ambos conjuntos, ocasionando una condición no deseable porque pierde su generalidad, y es conocida como ***sobreajuste*** o en inglés como ***overfitting*** [28].

Existen diversas técnicas para evitar el *sobre entrenamiento* mencionado, y además permiten obtener modelos funcionales cuando N_Λ no es numeroso. Algunos de estas herramientas son; la ***regularización*** de magnitudes de $\mathcal{W}$ y $\mathcal{B}$, variación dinámica de las muestras que conforman (3.31) y (3.38), método denominado ***k-fold*** o la desactivación parcial de ciertas neuronas de cada capa, conocido como ***dropout***, entre otras [28, 76, 78, 79, 81].

Con esto finaliza la síntesis de algunos conceptos clave de *redes neuronales*, cuyo propósito fue introducir términos relacionados para contar con un marco de referencia, que permita describir el uso de estos modelos de *aprendizaje automático* como descriptores de imágenes en la siguiente Sección.

Extracción de características por CNN

Una Red Neuronal Convolucional CNN es una arquitectura *feedforward* de múltiples capas, diseñada específicamente para reconocer características de arreglos bidimensionales, que en conjunto forman patrones con base en dependencias espaciales existentes en regiones locales, distribuidas en la señal de entrada [78].

Como su nombre lo indica, la operación fundamental de la red es la convolución [4] (3.1), que *actúa en cada neurona como un filtro de imagen*, donde los *pesos sinápticos* estructuran el núcleo. La respuesta de la neurona $\mathcal{y}_{\ell,d}$ es un arreglo 2D con un tamaño determinado por (3.2), cuyos elementos son cada uno de los resultados del filtro, acotados generalmente por la función de activación *ReLu* (3.28d).

La arquitectura de una red neuronal es considerada como *convolucional*, si alguna sección de su estructura esta formada por el ensamble de las *capas características* de una CNN [28]:

• ***Convolucional***; es un bloque en la red formado por un conjunto de neuronas, que aplican *filtros* a la señal de entrada por medio del operador (3.1). En los diagramas de arquitecturas esta capa se denotada comúnmente por la abreviación ***conv***, acompañada del tamaño y cantidad de *núcleos* usados en este bloque como $n_w \times n_w \times n_f$.

• ***Submuestreo***; la tarea de esta capa denominada en inglés como ***pooling*** y abreviada ***pool***, es reducir la dimensión del arreglo que se obtiene de una capa ***conv***, en caso de requerirse. Para la disminución, se divide cada arreglo del mapa en regiones cuadradas de 2×2 o 3×3 elementos sin traslape. De cada vecindario se determina un componente del nuevo mapa, que corresponde al promedio de la ventana ***average pooling*** o al elemento de mayor valor ***max-pooling*** que es el criterio más usual.

• ***Totalmente conectada***; es responsable de ajustar y procesar respuestas de una capa *conv* o *pool*, para determinar la respuesta final de la red $\mathcal{y}$ de acuerdo a su aplicación. En esta capa las neuronas son conectadas a todos los nodos de las precedentes o posteriores capas, como se ilustra en la Figura 3.1b, en donde también se indica su abreviación ***fc*** por su nombre en inglés ***full conection***.

Las combinaciones de ensamble de las capas *conv* y *pool*, forman el bloque de la red que funciona como un *descriptor de características*, por consiguiente sus respuestas son consideradas como *mapas de características*, que denotaré como $x_{\ell,\ell+1}$ donde ℓ indica la capa que lo genera y $\ell+1$ a la que entra. Las capas *fc* acomodan $x_{\ell,\ell+1}$ en un vector, el cual procesan para calcular la salida de la red.

Cuando una CNN es entrenada con imágenes de uno o varios objetos, las primeras capas *conv* tienden a describir elementos estructurales, como los bordes. Al procesar estos *mapas de características* en las capas subsecuentes de la red, los *pesos sinápticos* de cada bloque tienden

[4] Involucrando el desplazamiento del núcleo $\mathbb{s}$ y el posible *submuestreo* (3.2)

a ser más específicos, de manera que generan filtros de patrones locales singulares de los objetivos.

La **capacidad descriptiva** de la red depende del *entrenamiento* y de la cantidad de *neuronas* en las capas, porque definen el *espacio de características* principalmente determinado por los pesos sinápticos de las *capas ocultas*, integradas por combinaciones de bloques *conv* y *pool*.

Otro de los factores trascendentes para obtener una CNN funcional, es el nivel representativo del conjunto de observaciones utilizadas para el proceso de *aprendizaje*, porque deben incluir la mayor cantidad de variaciones y formas en las que se pueden presentar el o los objetos a identificar. Por ello dependiendo de los objetivos, el numero de ejemplares para entrenar puede requerir ser muy extenso.

Por esta razón, diferentes grupos de investigación han elaborado grandes compendios de imágenes, denominados en la literatura como ***datasets***, que contienen diferentes clases de objetos y anotaciones útiles para probar o entrenar algoritmos de clasificación y detección. Una de estas recopilaciones es *ImageNet* [13], famosa por ser el material de trabajo del reto ILSVRC, que consiste en identificar 1000 clases de objetos con el menor error posible.

Al no existir lineamientos o guias concretas para diseñar *arquitecturas* de CNN, los modelos que logran buenos resultados en pruebas como ILSVRC, son tomados en cuenta como estructuras de referencia, y que pueden adaptarse a diferentes aplicaciones.

Una de estas *arquitecturas* es la denominada *AlexNet* [20], mostrada en la Figura 3.4 reconocida en 2012 por impulsar el uso de redes neuronales para la tarea de *clasificación* al ganar la prueba ILSVRC del mismo año. Este hecho, la llevo a ser objeto de estudio de muchos trabajos de los cuales en [22], se demostró que la estructura sin su última capa fc_8, podía funcionar como un ***descriptor de características***.

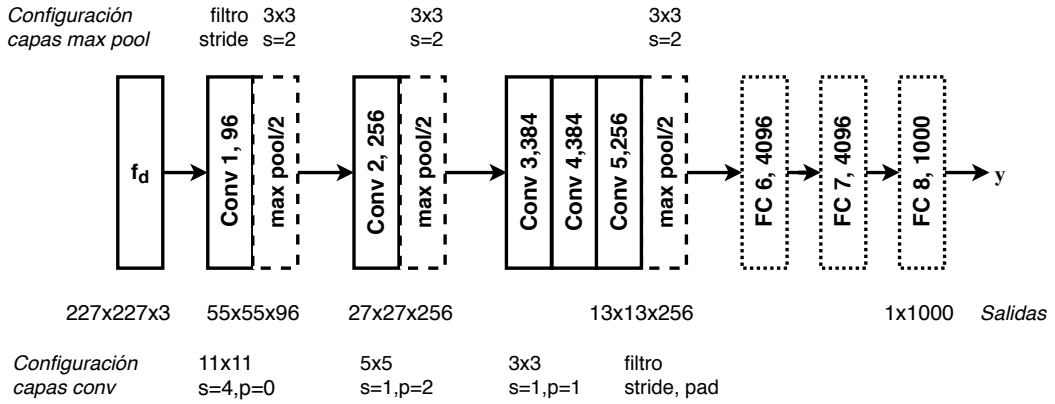


Figura 3.4. Arquitectura de la CNN AlexNet [20].

Aplicando las expresiones (3.2), se calculan las dimensiones del arreglo que genera cada una de las capas *conv* y *maxpool*, señalado en el diagrama de la arquitectura como *salidas*.

Otra *arquitectura* destacada como *descriptor de características* es el *bloque convolucional* de las diferentes versiones de la red denominada *VGG* [82], propuesta en ILSVRC del 2014, quedando en segundo lugar. Su versión mínima de 16 *capas*, mostrada en la Figura 3.5, logra crear un espacio de *características* más representativo que *AlexNet*, de acuerdo con posteriores sistemas de reconocimiento que la utilizaron como base de implementación, pero involucra una mayor cantidad de parámetros para ajustar en su entrenamiento.

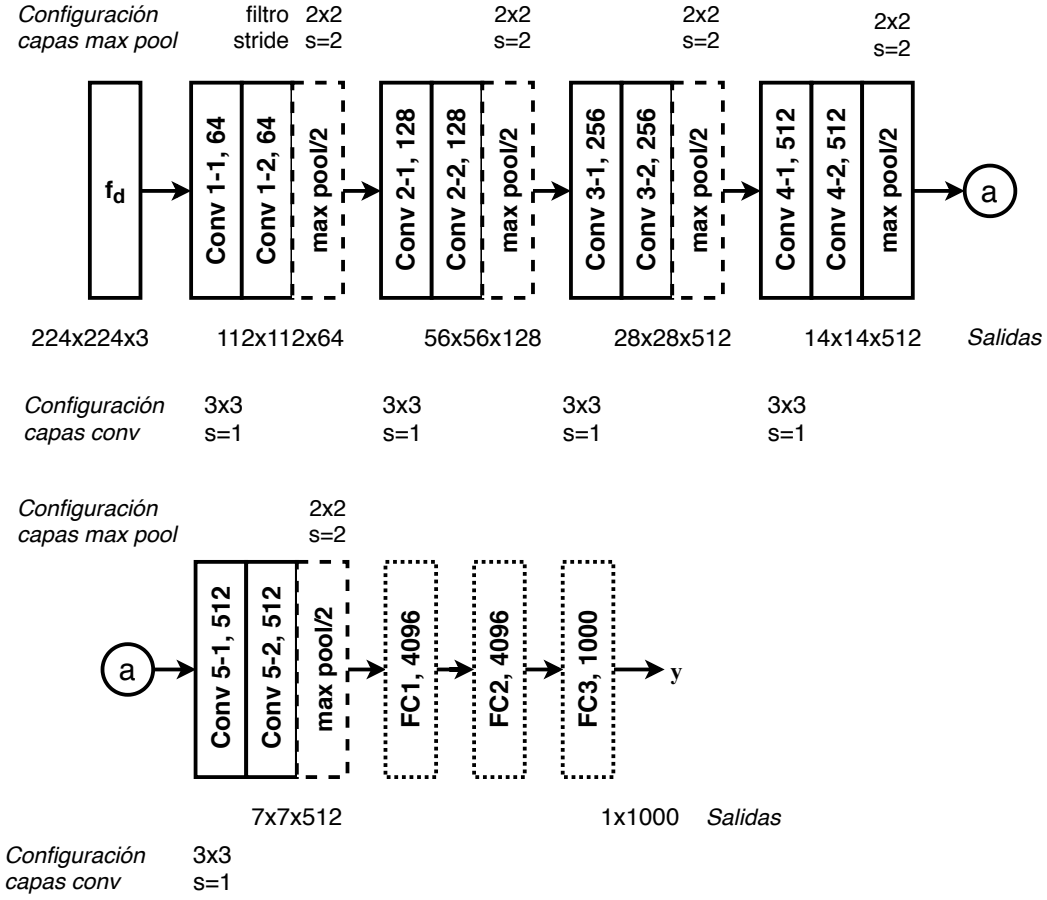


Figura 3.5. Arquitectura de la CNN VGG16 [82].

Ambas redes *aprenden* utilizan en su última capa una función de activación, que les permite distinguir uno de los diferentes patrones por cada neurona, de los objetivos de clasificación formados por distribuciones de características, que se extraen al filtrar la señal de entrada por $\mathcal{W}$ y $\mathcal{B}$ de las capas antecesoras.

Para catalogar dos clases, la última capa de una red puede consistir de una neurona con la función de activación (3.28c), y para más distribuciones categóricas se aplica la función ***softmax***, definida para una neurona $d = j$ como

$$y_{\ell,d}\big|_{d=j} = \Upsilon_s\left(z_{\ell,j}\right) = \frac{\exp\left(z_{\ell,j}\right)}{\sum_{d=1}^{n_\ell} \exp\left(z_{\ell,d}\right)} \tag{3.40}$$

en donde los subíndices ℓ, d señalan la capa y neurona evaluada. Como se observa en la expresión (3.40), esta función de activación depende de las activaciones de las otras neuronas en la capa,

para que la suma de todas las respuestas sea igual a uno [77].

Las *arquitecturas* CNN que se han posicionado en los primeros lugares del reto ILSVRC, después del 2012, la mayoría tiene una cantidad considerable de *capas*, razón por la que son denominadas redes de ***aprendizaje profundo***. Este factor común involucra, una demanda significativa de recursos computacionales y de conjuntos extensos de imágenes para *entrenar* los modelos.

Por esta razón, los autores de arquitecturas CNN hacen públicos sus programas computacionales de implementación, en diferentes bibliotecas de como *Keras* o *TensorFlow*, con el propósito de que otros desarrolladores puedan utilizar la *base de conocimiento* de la red, para hacerla especifica de acuerdo al objetivo de aplicación de cada proyecto. Esta práctica es denominada en la literatura como ***transfer learning***.

Aprovechando la capacidad descriptiva del contenido de imágenes, que han demostrado tener diferentes CNN, se han formulado *metodologías de clasificación e identificación* que logran hacer frente a diversos factores significativos en estas tareas, como son la observación parcial de un objetivo o la diversidad de formas en la que este se puede encontrar. Al ser afines algunas de estas circunstancias, en el siguiente Capítulo se mencionarán y describirán algunos de ellos, para efectos de su implementación.

Resumen

En este Capítulo se presentaron las metodologías y conceptos empleados para *describir formas humanas* mediante la generación de *modelos matemáticos* que tienden a ser constantes en el contenido de ciertas imágenes.

Las unidades fundamentales de los patrones de referencia son las *características*, definidas en espacios diferentes al de visualización gráfica a través de la aplicación de filtros, implementados por medio de la operación convolución, como se describió en la Sección 3.1.

Para el caso especifico del proyecto, en la Sección 3.2 se presentaron algunos *núcleos* de filtros, que detectan o resaltan rasgos bidimensionales representativos de formas antropomórficas, es decir características, utilizando transformaciones matemáticas diferenciales y funciones estadísticas.

La combinación de diferentes *filtros* detectores de características, generan dominios de características más robustos a diversas variaciones y deformaciones de los objetivos. A partir de estos, se generan los *patrones de referencia* aplicando algún algoritmo *descriptor*, como los expuestos en la Sección 3.3, que particularmente fueron seleccionados por emplearse para *describir formas humanas*, en diferentes proyectos antecedentes afines a la detección de víctimas.

En el siguiente Capítulo, se aborda la segunda fase de un algoritmo de clasificación y/o detección, que es la comparación del modelo de referencia utilizando planteamientos de verosimilitud.

Capítulo 4

Algoritmos para detectar personas

Las proyecciones de los objetos a reconocer en una imagen, están formados por conjuntos de pixeles definidos convencionalmente en alguno de los dominios de percepción gráfica. Estos datos son transformados a otro dominio por algún *descriptor* de *características*, y mediante diferentes tipos de criterios y modelos de distribución, generales o específicos determinados por un proceso de *aprendizaje*, el dominio de *características* estructura un conjunto de *patrones*, que corresponden a los diferentes objetos que se desean reconocer.

En este Capítulo se presentan algunos de los algoritmos de ***clasificación y detección de objetos*** orientados a personas, reportados en trabajos antecedentes afines, con la finalidad de plantear una metodología que aproveche el campo de visión compartido de las cámaras IR y VIS.

En la Sección 4.1 se expondrá el contexto de la tarea de clasificación y detección de objetos, incluyendo una reseña breve de los métodos que existen. Posteriormente, de acuerdo al planteamiento del proyecto, se describen formas más especializadas de análisis del contenido de una imagen, que permiten reconocer los objetivos sin la necesidad de tener que visualizarlo por completo. Y en la Sección 4.3 se explican diferentes indicadores que permiten evaluar el rendimiento y funcionalidad de los algoritmos que reconocen y localizan objetos.

4.1 Clasificación y detección de objetos

Respecto al campo de visión por computadora, esta tarea consiste en ***identificar*** uno o más objetos específicos proyectados en imágenes, analizando su contenido con métodos que determinan una ***afinidad*** entre, colecciones de pixeles de una imagen o región de esta f_d y α ejemplares de los objetivos a reconocer, denominados comúnmente como *patrones*.

Esta aplicación se fundamenta en los algoritmos de *aprendizaje automático* denominados ***clasificadores***, los cuales aprenden a relacionar señales del mismo tipo, con un conjunto finito de respuestas denominadas ***clases***, mediante la formación de un modelo de estimación $\mathcal{Y}' \epsilon \mathbb{R}^{\alpha}$, de la función que define las asociaciones de entrada-salida requeridas $\mathcal{Y} = \mathrm{F}(x)$ [79].

El procedimiento básico de un sistema de *clasificación*, es estructurado comúnmente en las etapas indicadas en la Figura 4.1, personalizado el diagrama en atención al enfoque del proyecto, sin perder su carácter general para otras aplicaciones.

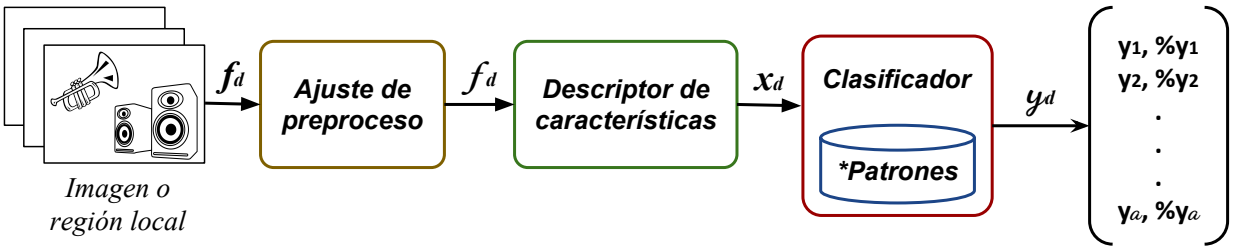


Figura 4.1. Diagrama de bloques del sistema de clasificación de objetos.

La fase *Ajuste de preproceso* consiste de un conjunto de operaciones, que preparan y/o acondicionan la señal de entrada, en caso de ser necesario, para extraer representaciones distintivas aplicando un *descriptor de características* en la segunda etapa, con la finalidad de facilitar su análisis en la siguiente parte del proceso.

De acuerdo al ***mapa de características*** x_d, se aplica una metodología de *clasificación*, formada por uno o varios criterios que asocian la señal de entrada f_d con alguna de las posibles salidas $\mathcal{Y}_d$. La técnica para catalogar puede componerse de uno o diferentes tipos de normas, considerando como base medidas de carácter geométrico, valoraciones lógicas o estimaciones de probabilidades que definen el tipo respuesta $\mathcal{Y}'_d$, de los clasificadores base mostrados en la Figura 4.2.

La mayoría de las aplicaciones de *reconocimiento de objetos* en imágenes, tienen como principio de su planteamiento, alguno de los métodos de *aprendizaje supervisado* mencionados en la Figura 4.2. Como algunos de estos *clasificadores* son parte de sistemas dedicados y afines al *reconocimiento de víctimas*, a continuación se presenta una descripción breve ellos, con la finalidad de establecer un contexto básico para la Sección 4.2.

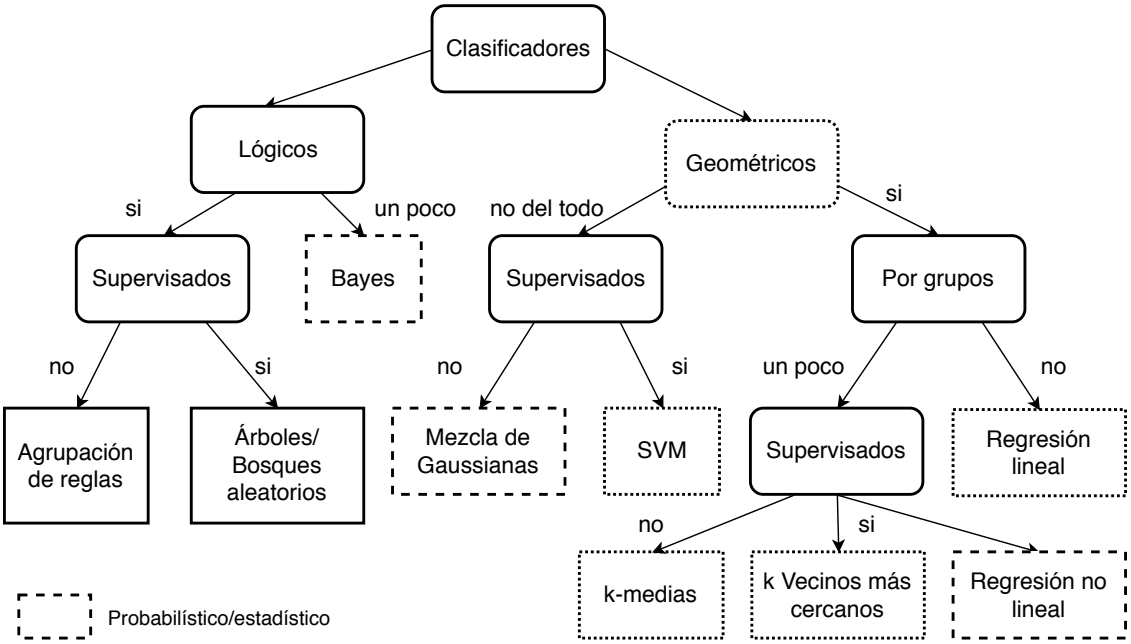


Figura 4.2. Métodos de clasificación base, ordenados por su tipo de aprendizaje y tipo de principio de catalogación [83]. Los rectángulos sin esquinas redondeadas, corresponden a los algoritmos y el tipo de línea al criterio base que consideran.

• ***Árboles de decisión***; cataloga observaciones mediante un análisis estructurado en diferentes niveles, representados por *grafos* dirigidos, en donde se evalúan los criterios de diferenciación, costos de decisión y otros parámetros que determinan la clase de salida [83].

Aunque el método esta asociado a un carácter lógico, la respuesta final puede ser expresada como probabilidades de semejanza o puntuaciones de error. Y adicionalmente si las señales a clasificar lo permiten y requieren, la salida del algoritmo puede ser producto de más de un árbol de diferente tipo, denominando a estas combinaciones como ***bosques aleatorios*** [84].

Algunos ejemplos de aplicación vinculados directamente con el reconocimiento de objetos en imágenes son; el algoritmo *Adaboost* de *filtros Haar* en cascada, propuesto por Viola y Jones en [4], el método DPM [10] o el caso particular de las *redes neuronales convolucionales* [76].

• ***k-Vecinos más Cercanos, k-NN***; este método supone que las α categorías objetivo, son suficientemente definidas en el *espacio de características* asociado con el *descriptor*, por lo que designa alguna clase por el criterio de *cercanía geométrica*.

Considerando τ ejemplares de cada categoría en el dominio de x, una observación de entrada f_d es clasificada a la clase α, si la distancia entre k de sus ejemplares y f_d es menor que las otras categorías. La magnitud de adyacencia es calculada por alguna función que mida la separación entre dos elementos, como la *norma Euclidiana* [85].

A diferencia de otros procedimientos, este algoritmo solo requiere que se especifiquen los $\alpha \cdot \tau$ patrones, para implementarse. Algunas de sus aplicaciones son; probar descriptores y reconocer objetos rígidos poco variantes, como señalamientos de tránsito [83].

• ***Máquina de Soporte Vectorial, SVM***; es un algoritmo que diferencia solo dos clases, porque considera que el espacio de características de dimensión n, puede ser seccionado en dos por una frontera denominada ***hiper-plano***, especificado por la siguiente expresión

$$y_k = w_k{}^T \phi(f) + b_k = w_k{}^T x + b_k \quad | \quad y_k = \begin{cases} 1 & y_k > 0 \\ -1 & y_k < 0 \end{cases} \tag{4.1}$$

en donde ϕ representa el *descriptor* aplicado a las observaciones f, para clasificarlas en alguna de las dos respuestas y_k, en función de la posición del *hiper-plano* determinado por $w_k{}^T$ y b_k, que se definen por un proceso de *entrenamiento* inspeccionado [17].

El modelo (4.1) es conocido como ***Hard SVM***, porque logra clasificar señales, si sus transformaciones son linealmente separables en el espacio de x. Pero si no tienen esta tendencia, es posible integrar un ajuste a (4.1) para poderlas catalogar, versión que es denominada como ***Soft SVM*** [85].

Cada algoritmo de *clasificación*, tiene un proceso especifico para ajustar sus correspondientes parámetros de selección y formar los modelos de referencia necesarios para la aplicación. Sin embargo, la mayoría de las metodologías de adecuación referidas como ***aprendizaje***, tienen dos principios de llevarse a cabo; con supervisión o sin ella.

Ambos modos de *capacitación* se describieron en la Sección 3.3.4, enfatizando la modalidad de *entrenamiento supervisado*, porque el *descriptor* de características involucrado en las CNN, se define junto con los criterios de *clasificación* de la red. Excluyendo las técnicas de *optimización* del gradiente descendente y la función de pérdida especifica (3.32), el resto del desarrollo es característico de cualquier método de aprendizaje automático inspeccionado [79, 80].

Los ejemplares de entrenamiento (3.31) para un *clasificador* de imágenes, deben ser preferentemente proyecciones que solo contengan los objetos a reconocer, para que los parámetros de selección sean exclusivos. Por esta razón, el método no es capaz de indicar donde se encontró alguno de los objetivos.

Sin embargo, en aplicaciones no controladas los objetos de interés no están solos en la escena, por ello para localizarlos convencionalmente se aplica una ***estrategia de búsqueda***, que proporcione regiones f_d de cada imagen, para procesarse por el esquema de la Figura 4.1.

Una de las técnicas más convencionales es la ***ventana deslizante***, que determina regiones de la misma forma que se desplaza el núcleo de una convolución 2D. Al no ser muy eficiente esa formulación, se han propuesto tácticas que seleccionan bounding boxes a partir de diferencias de contraste o por la combinación de dichos cambios con algunas características como en [24].

Al incluir este paso, el sistema de reconocimiento se convierte en un ***detector de objetos***. De acuerdo a la naturaleza de los objetivos a localizar, se define el método más adecuado para conseguir hacer la tarea, en la siguiente Sección se describirán los algoritmos que probablemente puedan encontrar víctimas con oclusiones parciales.

4.2 Detección fraccionada de objetos

Al plantear un sistema de detección o clasificación de objetos, uno de los principales factores a considerar en su formulación, son los estados en los que se pueden presentar los objetivos en las imágenes, entendiéndose por ello; sus factibles posturas que determinan la cantidad de siluetas que se proyectan, las circunstancias del entorno en el que se encuentra inmerso, las diferentes texturas de las superficies que lo componen, entre otras condiciones.

La lista de elementos a tomar en cuenta, se define de acuerdo a las delimitaciones de diseño y la *fisonomía* de los objetos. Para el caso de interés de este proyecto, los trabajos sobre la detección de cuerpos con articulaciones han desarrollado un principio de *análisis fraccionario*.

Esta estrategia de estudio es considerada como la más adecuada para la *detección de víctimas* en entornos post desastre, porque de acuerdo a la documentación consultada, es capaz de afrontar algunas de las dificultades listadas en la Sección 1.2. Por esta razón, a continuación se presenta una breve reseña de los algoritmos factibles para la implementación del proyecto.

4.2.1 Modelado de objetos con piezas deformables, DPM

Como su nombre lo expresa, es una metodología de análisis para describir y reconocer cuerpos, seccionándolos en una estructura gráfica que representa las partes significativas de su fisonomía, que se tendrán que detectar de forma parcial o total para lograr la tarea.

A pesar de que el enfoque de dividir la tarea no era una novedad, Fischler y Elshlager en [11] fueron quienes plantearon una de las técnicas que más se ha desarrollado para el reconocimiento de objetos con este principio, denominada como ***Pictorial Structures PS***, que consiste en describir un cuerpo por una colección de partes, conectadas de acuerdo a la estructura de un *grafo no dirigido*.

El planteamiento supone que un cuerpo B esta compuesto por p piezas, que están conectadas de acuerdo con a la complexión de un esqueleto de referencia $\mathcal{S}$, especificado por las posiciones relativas de cada nodo del grafo como

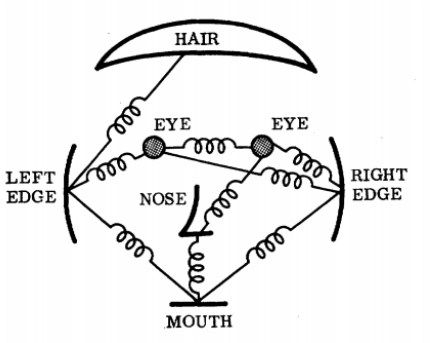
$$\mathrm{B} = \{\rho_k\}_{k=1}^{p} \quad k\epsilon\mathbb{N} \mid k = 1, 2, \dots, p \qquad (4.2a) \qquad\qquad \mathcal{S} = \{s_k\}_{k=1}^{p} = \{(i_k, j_k)\}_{k=1}^{p} \qquad (4.2b)$$

en donde ρ_k representa cada componente de B, el par de coordenadas s_k corresponde al punto central donde se ubica cada parte ρ_k, considerando como origen del sistema de referencia a s_1, porque ρ_1 es el *nodo de referencia* para analizar el grafo. El método PS plantea una *función costo*, que cuantifica cuanto se debe ajustar el esqueleto (4.2b), para coincidir con una región

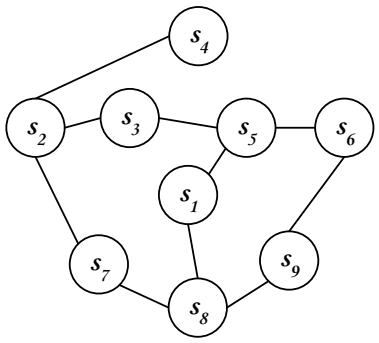
que posiblemente tenga una proyección del objetivo [11, 12]. De manera que, el esfuerzo total de adaptación es mesurado como

$$\mathcal{E} = \sum_{k=1}^{p} \sum_{m=1}^{k} l\,(s_k, s_m) \tag{4.2c}$$

donde l es una función que mide la relación espacial geométrica entre dos partes conectadas, representando la *deformación* de su vinculación. Por esta razón las relaciones geométricas entre las partes del objeto, se interpretan como un resorte entre las diferentes posiciones características de cada pieza, estructuradas de acuerdo a un grafo, como se ilustra en la Figura 4.3.



(a) Representación de un rostro por con una estructura pictórica.



(b) Grafo no dirigido propuesto para describir el rostro de la Figura 4.3a.

Figura 4.3. Ilustración de la implementación de la descripción esquemática PS [11].

Para clasificar o detectar objetos, a partir de parámetros iniciales, se implementa un proceso de *aprendizaje* para crear el ***modelo generativo*** (4.2b) patrón y entrenar los *clasificadores* dedicados a cada parte de B, mediante la optimización de la siguiente expresión

$$\mathcal{E}^* = \arg\min_{\mathcal{E}} \left(\sum_{k=1}^{p} e\,(\rho_k) + \mathcal{E} \right) \tag{4.2d}$$

en la cual introduce el error de detección total de las piezas $e\,(\rho_k)$, definiendo el esquema de descripción estructurado combinado con una colección de clasificadores, cuyo desarrollo fue principalmente a nivel teórico.

En el año 2005, Felzenszwalb y Huttenlocher publicaron en [12] una metodología para reconocer personas de cuerpo completo y rostros, formulación que destacó por implementar el concepto de PS y establecer una base teórica vinculada a la práctica, que impulso el modelado de objetos con partes deformables.

Desarrolló su propuesta inicial hasta lograr definir un algoritmo general para *detectar objetos en múltiples escalas*, con la capacidad de afrontar oclusiones parciales y variaciones de pose por articulaciones [9, 10].

En este método *la estructura de un objeto*, se restringe a **grafos dirigidos acíclicos tipo árbol**, cuyo *nodo raíz* ρ_1, define la posición relativa de las partes del cuerpo (4.2a), especificando cada

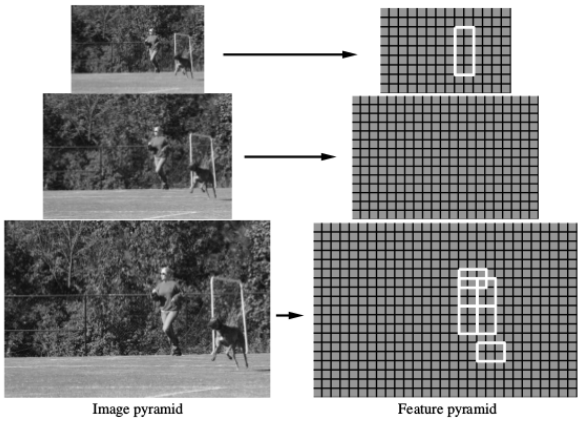
una de estas piezas como

$$\rho_k = (F_k, v_k, s_k, a_k, b_k)^T \tag{4.3}$$

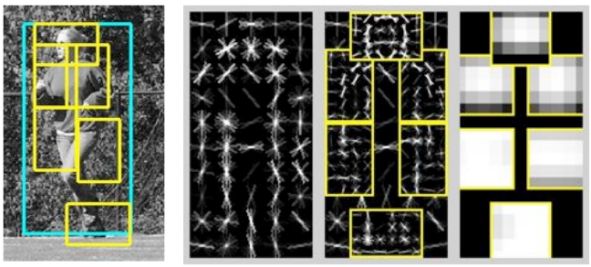
en donde F_k es el núcleo del filtro aplicado a cada ρ_k, de $w_{f_k} \times h_{f_k} \times 9 \times 4$ coeficientes de la región que contiene la parte k, especificando su punto central como $v_k = (x_k, y_k)$ y las longitudes del bounding box por $s_k = (w_b, h_b)$. Y los vectores $a_k = (a_1, a_2)|_k$ y $b_k = (b_1, b_2)|_k$, son los coeficientes de una función cuadrática, usada para evaluar la ubicación de la pieza k.

Para detectar un objeto analiza la estructura modelo en diferentes escalas, para lo cual genera una *pirámide de octavas* en donde cada nivel es un *mapa de características* HOG x_ℓ, de toda la imagen como se muestra en la Figura 4.4a. definiendo cada vector HOG de x_ℓ con el Algoritmo 3.3.1.

En los niveles de menor resolución, busca detectar la parte que corresponde al nodo raíz aplicando la técnica de *ventana deslizante*. Al encontrarla, comienza a analizar el resto de la estructura en niveles de mayor resolución a partir de la localización de la parte ρ_1. Esta táctica de búsqueda, agilizaba la detección y permite examinar detalladamente la fisonomía de cada parte del modelo, como se observa en la Figura 4.4b



(a) Ilustración de las pirámides mapas de características HOG [10].



(b) Análisis y detección de componentes que forman el modelo B, en el espacio determinado por el descriptor [9].

Figura 4.4. Representaciones que ejemplifican la búsqueda de un objeto, en diferentes escalas usando una pirámide de octavas y sus diferentes resoluciones, para buscar y analizar el modelo estructurado DPM.

Para integrar los parámetros relacionados al análisis de B y de los *clasificadores* dedicados a cada ρ_k, utilizó una SVM latente, organizando todas las variables discriminativas para ajustarlas en un solo *proceso de aprendizaje*, definiendo la calificación de detección del objeto por la siguiente expresión

$$\mathrm{B}_\omega = max_{\mathcal{S}} \left(\omega \cdot \phi \left(\mathbb{H}_f, \mathcal{S}\right)\right) \tag{4.4}$$

en donde ω es el arreglo de parámetros que definen el modelo, formado

$$\omega = \left\{F_0, \ldots, F_p, a_1, b_1, \ldots, a_p, b_p\right\}^T \tag{4.5}$$

y las posiciones de referencia de cada una de las partes que forman la estructura del cuerpo, están señaladas por

$$\phi\left(\mathbb{H}_f, \mathcal{S}\right) \tag{4.6}$$

donde $\mathbb{H}_f$ indica la pirámide de mapas de características, generada para analizar una imagen o región de local de esta f. Y $\mathcal{S}$ es el conjunto de posiciones de cada ρ_k, como en (4.2b) pero extendiendo el arreglo como $s_k = (i_k, j_k, \ell)^T$ para especificar el nivel [10].

Otra de las implementaciones prácticas de PS que resaltaron, fue el planteamiento de Andriluka fundamentado en la revisión teórica [12]. Su línea de investigación tuvo por objetivo formular un detector de personas capaz de afrontar oclusiones, cambios de iluminación y variaciones de los cuerpos, debido a diferentes posturas de sus articulaciones.

A diferencia de Felzenszwalb, su metodología conservó más el fundamento probabilístico [12] al que integró un conjunto de clasificadores Adaboost, entrenados para cada parte del objeto usando el descriptor *Shape Context* [66], el cual obtiene un histograma de orientación de gradientes de toda la región de imagen[1].

Posteriormente, Andriluka comparó el desempeño de su planteamiento con otras metodologías, incluyendo el esquema DPM de [10], en una prueba que consistió en detectar personas ocluidas, simulando ser víctimas en un escenario de búsqueda y rescate urbano [7].

En este experimento, observó que la combinación de los métodos [8, 10], lograba obtener una precisión de detección superior a la que obtuvo cada uno, y por esta razón concluyó que la detección objetos en circunstancias complejas, podría llevarse a cabo modelando los objetivos en partes deformables.

4.2.2 Red neuronal convolucional regional, R-CNN

Es un planteamiento para detectar objetos en imágenes propuesto por Ross Girshick en [29], capaz de afrontar oclusiones y diferentes estados de los objetivos, sin realizar un análisis directo de las partes que lo componen, como en el caso de DPM. En la Figura 4.5 se muestra un diagrama de los tres módulos que componen este procedimiento.

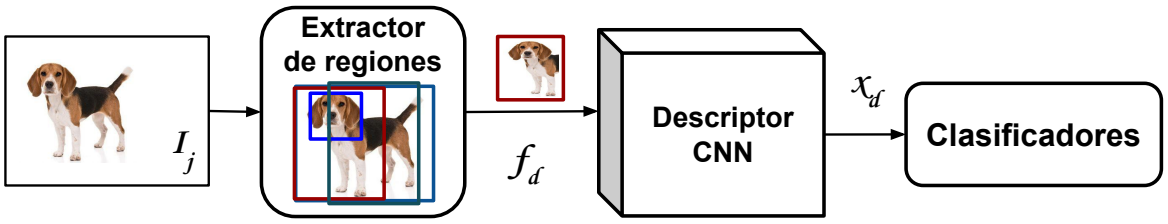


Figura 4.5. Diagrama de bloques de la red neuronal convolucional regional.

La palabra *regional* hace referencia al núcleo de la metodología, porque implementa una estrategia diferente al paradigma de *ventana deslizante*, para buscar regiones que probablemente contengan

[1]Dicho histograma se forma como el descriptor local de un punto SIFT, explicado en el Algoritmo 3.3.2

algún objetivo de detección. Y adicionalmente, a partir de un conjunto de entrenamiento con anotaciones de bounding box, denotado como T

$$\mathrm{T} = \left\{\left(\mathrm{I}_j, \beta_j, \mathcal{Y}_j\right)\right\}_{j=1}^{N_\mathrm{T}} \qquad (4.7a) \qquad \beta_j = (x, y, w, h)^T \qquad (4.7b)$$

genera observaciones parciales

$$\tau_{j,d} \subset \mathrm{I}_j | \tau_{j,d} = \left\{\left(f_{j,d}, b_d, \mathcal{Y}_j\right)\right\}_{d=1}^{N_\tau} \qquad (4.7c)$$

de cada proyección del objeto $f_{j,d}$, con regiones que incluyen diferentes partes del área delimitada por β_j, aplicando la misma táctica de búsqueda inicial, con ajustes específicos para obtener ejemplares positivos y negativos, de acuerdo a un porcentaje de intersección. Los vectores β_j y b_j se componen por las coordenadas del punto superior izquierdo del bounding box, seguido del ancho y largo del cuadrilítero.

De esta manera, el descriptor y los clasificadores ajustan sus parámetros para poder responder a oclusiones de un objetivo, sin involucrar un análisis especifico de su fisonomía. Brevemente, el procedimiento de la Figura 4.5 consiste de las siguientes acciones:

El módulo ***Extractor de regiones***, corresponde a la estrategia de *búsqueda regional* que consiste del método *Búsqueda Selectiva* [24], el cual obtiene una lista de regiones $f_{j,d} \subset \mathrm{I}_j$ que probablemente tengan de forma parcial o total un objeto. Está técnica define bounding boxes a partir de un conjunto de medidas de similitud, determinadas por perfiles de color, áreas de regiones de pixeles con valores casi homogéneos, análisis de texturas entre otros.

Posteriormente, cada región $f_{j,d}$ es ajustada en tamaño para ingresar al ***Descriptor* CNN**, que corresponde a la arquitectura *Alexnet* mostrada en la Figura 3.4, sin la capa fc_8, obteniendo un vector de características $x_{j,d}$ de 4096 elementos. La CNN fue entrenada en dos etapas; primero de forma general con el *dataset ImageNet* [20] y después manera especifica usando las observaciones $\tau_{j,d}$.

Y el último bloque del método R-CNN, denominado como ***Clasificadores*** esta formado por α SVM, entrenadas cada una para uno de los objetos a reconocer, para que sea capaz de distinguir un objetivo al evaluar $x_{j,d}$.

En la publicación de este método [29], Girshick compara el rendimiento de su propuesta con otros algoritmos, incluyendo el planteamiento de Felzenszwlab [10], en el cual colaboró. De este análisis de desempeño utilizando *datasets* canónicos, declaró que el espacio de características que se define en una CNN, manifiesta una mayor capacidad representativa y selectiva, por lo que al usarla como *descriptor* con el concepto de *regiones propuestas*, es más eficiente que DPM.

Esta afirmación fue el fundamento de desarrollo, de otras metodologías para *detección de objetos* en circunstancias poco controladas. Algunos de estos nuevos sistemas, lograron superar el 5.1 % de error de detección de objetos humano, estimado para la clasificación de ILSVRC, aplicando las siguientes contribuciones:

1 *Fast R-CNN* [86]; considerada como la segunda versión de [29], en donde cambió el descriptor AlexNet por VGG16, del cual definía un x_d que lo hacia vector con una capa de *agrupamiento* llamada *roi pooling*. También sustituyó los clasificadores SVM por una capa fc e incorporó en paralelo a esta capa de clasificación, otra para estimar bounding boxes.

2 *Faster R-CNN* [87]; buscó agilizar el entrenamiento y reducir el tiempo de procesamiento, para poder implementar un detector en línea. Tomando como base Fast R-CNN, se sustituyó el método para proponer regiones por una CNN denominada *Regional Proposal Network*.

4.2.3 Modelado de objetos con partes deformables usando CNN

Después de que las redes neuronales convolucionales, fueron consideras como una de las mejores herramientas para *describir* y/o *clasificar* imágenes, por los resultados mostrados por AlexNet, algunos desarrolladores comenzaron a trabajar con DPM utilizando como *descriptor* de características alguna CNN o realizando combinaciones de varios extractores de singularidades, entre algunas otras ideas.

Ouyang en [27], propuso una arquitectura de CNN para detectar peatones, adaptando el concepto de análisis estructural de un DPM, mediante una mapa de detección de partes que denominó *capa deformable*, como se observa en el esquema de la Figura 4.6.

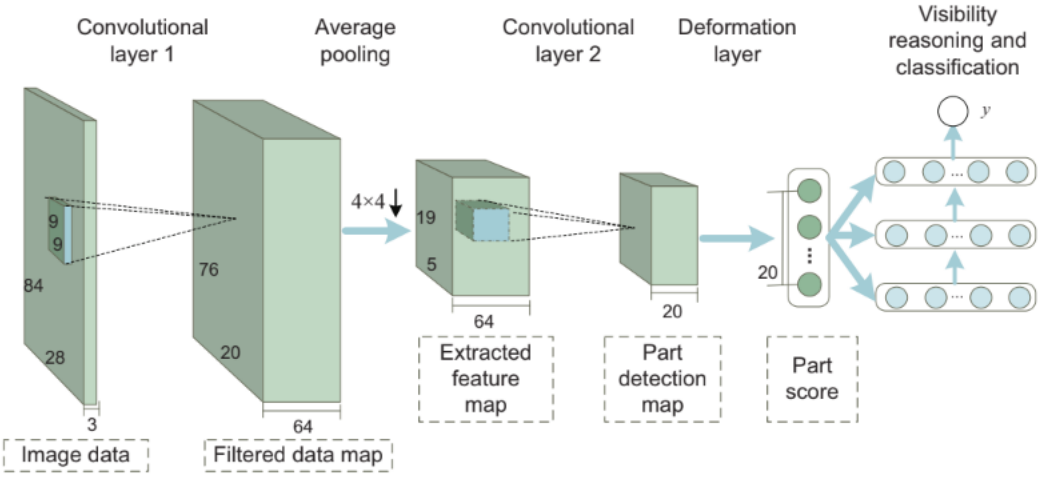


Figura 4.6. Arquitectura de la CNN propuesta para implementar un DPM [27].

La señal que ingresa a esta red, es de particular interés para los fines de este proyecto, porque a pesar de trabajar también con imágenes en el espacio RGB, optó por convertir sus imágenes al espacio YCbCr, bajo el argumento de afrontar mejor los cambios de contraste, debidos a variaciones de iluminación. El tensor que utiliza para detectar peatones, de 84×28 esta formado por estos canales:

1 El canal de luminiscencia Y.

2 Un mapa de intensidades de pixeles integrado por cuatro secciones; cada una es un canal de la conversión YCbCr de la región de interés escalada y ajustada a 42×14 pixeles. La restante es completada con una matriz de ceros.

3 Un mapa de bordes, integrado por cuatro secciones de 42×14 pixeles. Tres de estas secciones son los bordes calculados de cada canal Y, Cb y Cr y el cuarto mapa se forma de los máximos registrados, elemento por elemento de cada uno de los mapas anteriores.

Posteriormente Girshick publicó en [88], una metodología general para implementar la técnica DPM, en una CNN adaptada para trabajar en múltiples escalas, con el objetivo de demostrar que el análisis de ambas herramientas para detectar objetos, eran en cierta medida equivalentes, aunque con la estrategia de R-CNN se podían lograr mejores resultados, que en la actualidad fueron superados por otros sistemas propuestos.

4.3 Evaluación estadística de decisiones

Los algoritmos presentados en la Sección 4.2, catalogan observaciones delimitadas por *bounding boxes*, de acuerdo a grafos estructurados definidos por modelos probabilísticos o redes neuronales artificiales. Cada uno de estos métodos, utiliza una medida de error durante su proceso de *aprendizaje supervisado* para ajustar sus criterios de clasificación.

Sin embargo, los procesos de entrenamiento y evaluación de los algoritmos de aprendizaje automático, también son inspeccionados por un conjunto de indicadores estadísticos, que cuantifican la toma de decisiones del método [83]. Antes de presentar los índices de valuación, en la Tabla 4.1 se listan y describen el conjunto de términos involucrados en su definición.

Tabla 4.1: Términos para describir las respuestas de un algoritmo de clasificación, para evaluar su desempeño.

Término	Abreviación	Descripción
Verdaderos Positivos	TP	Notación empleada para señalar respuestas acertadas, en donde la clase estimada corresponde con la real.
Verdaderos negativos	TN	Asignación correcta de observaciones, al asociarlas con clases a las que no corresponden.
Falsos positivos	FP	Relación de clases asociadas por el modelo, a observaciones que no pertenecen a dichas clases.
Falsos negativos	FN	Vinculaciones de muestras de una clase, con predicciones incorrectas del modelo.

El grupo de símbolos de la Tabla 4.1, describen las posibles asociaciones de observaciones con las diferentes clases, que puede determinar un algoritmo de *clasificación* de forma correcta e incorrecta, al realizar una evaluación del modelo durante su entrenamiento y al finalizarlo, utilizando los conjuntos de observaciones de validación $\mathcal{V}$ y de evaluación final $\mathcal{E}$, para ver el nivel de generalidad conseguido.

Los resultados de una prueba son recopilados en un arreglo 2D denominado ***matriz de confusión*** o *tabla de contingencia* [83], formada de $\alpha \times \alpha$ elementos, organizados como se muestra en la Tabla 4.2. En los renglones de este arreglo, se especifican las α categorías conocidas, de las observaciones que serán procesadas por el modelo, para que esté les asigne una clase representada en las columnas como se muestra a continuación.

Tabla 4.2: Estructura de una matriz de confusión [83].

		Asociación estimada por el modelo					
		Clase 1	*Clase 2*	*Clase 3*	…	*Clase* α	*Renglón marginal*
Categoría verdadera de la observación	***Clase 1***	$y_{1,1}$	$y_{1,2}$	$y_{1,3}$	⋯	$y_{1,\alpha}$	$\sum y_{i=1,j}$
	Clase 2	$y_{2,1}$	$y_{2,2}$	$y_{2,3}$	⋯	$y_{2,\alpha}$	$\sum y_{i=2,j}$
	Clase 3	$y_{3,1}$	$y_{3,2}$	$y_{3,3}$	⋯	$y_{3,\alpha}$	$\sum y_{i=3,j}$
	⋮	⋮	⋮	⋮	…	⋮	…
	Clase α	$y_{\alpha,1}$	$y_{\alpha,2}$	$y_{\alpha,3}$	⋯	$y_{\alpha,\alpha}$	$\sum y_{i=\alpha,j}$
	Columna marginal	$\sum y_{i,j=1}$	$\sum y_{i,j=2}$	$\sum y_{i,j=3}$	⋯	$\sum y_{i,j=\alpha}$	$\sum y_{i=j}$

Cada uno de los elementos $y_{i,j}$ es la cantidad de observaciones asociadas por el algoritmo de clasificación, de acuerdo a la asociación del modelo y a la clase que pertenece, considerando las correspondientes intersecciones renglón-columna.

Del registro de resultados de la Tabla 4.2, cada uno de los elementos de la diagonal principal indican el número de *verdaderos positivos*, obtenidos en las diferentes clases disponibles

$$TP_c = y_{i,j|_{i=j=c}} \tag{4.8a}$$

Las asignaciones incorrectas de clase indicadas por el algoritmo, son respectivamente

$$FP_c = \sum_{j=1;j\neq c}^{\alpha} y_{j,c} \tag{4.8b}$$

y el caso opuesto es

$$FN_c = \sum_{j=1;j\neq c}^{\alpha} y_{c,j} \tag{4.8c}$$

en donde el subíndice c, indica en (4.8b) la clase estimada por el modelo y en (4.8c) la clase real de una observación. Y por último los *verdaderos negativos* de cada categoría c, son especificados como

$$TN_c = \sum_{j=1;j\neq c}^{\alpha} \sum_{k=1;k\neq c}^{\alpha} y_{j,k} \tag{4.8d}$$

Con base en la *matriz de confusión* y las posibles asociaciones (4.8), para un clasificador de $\alpha > 2$ categorías, se definen las siguientes mediciones que cuantifican el desempeño del modelo.

4.3.1 Exactitud

Referida en inglés como ***accuracy***, es un indicador que representa la proporción de observaciones de un total de $\mathcal{N}$ empleadas en una prueba de evaluación, que fueron relacionadas como *verdaderos positivos* con cada una de las respectivas clases

$$\mathcal{A}cc = \frac{1}{\mathcal{N}} \sum_{i=j=c}^{\alpha} y_{i,j} = \frac{1}{\mathcal{N}} \sum_{c=1}^{\alpha} TP_c \qquad | \quad c = 1, 2, \dots, \alpha \text{ y } \mathcal{N} = \{N_{\mathrm{T}} || N_{\mathcal{V}} || N_{\mathcal{E}}\} \tag{4.9a}$$

Cuando se utiliza un ***clasificador binario*** $\alpha = 2$, la exactitud se calcula considerando los *verdaderos negativos*, de manera que (4.9) se reescribe como

$$\mathcal{A}cc_b = \frac{TP + TN}{\mathcal{N}} \tag{4.9b}$$

La medida (4.9), es utilizada como un segundo indicador del progreso de *aprendizaje* de una CNN, además del error calculado por la *función de pérdida*, que guía el ajuste de los parámetros (3.29) y (3.30). Ambas mediciones se obtienen al finalizar un *epoch* de *entrenamiento*, usando para dicha prueba el conjunto de observaciones de validación $\mathcal{V}$ [81].

4.3.2 Precisión

Es el índice que cuantifica la fracción de observaciones asociadas adecuadamente por el *clasificador*, a su clase correspondiente tomando en cuenta todos los aciertos y vinculaciones a dicha clase de muestras que no pertenecen a ella [83]. Esta medida representa la *confianza* de obtener una asignación correcta, porque cuantifica la *variabilidad estadística* de vinculación, para la categoría c de la siguiente manera

$$\mathcal{P}_c = \frac{TP_c}{TP_c + FP_c} \tag{4.10}$$

4.3.3 Índice de certidumbre

Medida complementaria de la *precisión*, denominada en inglés como ***recall***, señala la proporción de observaciones de una categoría c que fueron asignadas correctamente por el algoritmo, con respecto a dichos aciertos y estimaciones erróneas de muestras pertenecientes a la clase en cuestión, con el resto de posibles respuestas [79]. En términos de la Tabla 4.1, el *recall* para una clase c es calculado como

$$\mathcal{R}_c = \frac{TP_c}{TP_c + FN_c} \tag{4.11}$$

Los indicadores (4.10) y (4.11) al ser determinados por cada clase disponible en el modelo, son considerados como mediciones binarias, porque suponen que las categorías restantes como una sola, representando asignaciones incorrectas, por esta razón son comúnmente asociadas solo con *clasificadores binarios*.

Sin embargo, al realizar una gráfica de un registro de los indicadores *recall* y *precisión*, se obtiene una visualización más descriptiva del desempeño del algoritmo, en la asignación de observaciones por cada una de las clases [79]. Esta representación gráfica se llama ***curva recall-precisión*** y

es ampliamente utilizada para comparar respuestas de clasificadores.

A partir de la curva *recall precisión*, se define otro indicador llamado ***precisión media***, abreviado **AP** en inglés, que equivale al *área bajo la curva* en cierto intervalo, referida como *AUC* en inglés. Una de las formas de cuantificar AP, es la determinada para VOC en [23], que consiste para una clase en la siguiente expresión para una clase es con la siguiente expresión

$$AP_c = \left(\mathcal{R}_{c,i} - \mathcal{R}_{c,j}\right) \cdot max\left(\mathcal{P}_c\left(\mathcal{R}_{i,j}\right)\right) \tag{4.12}$$

en donde $max\left(\mathcal{P}_c\left(\mathcal{R}_{i,j}\right)\right)$, es la máxima precisión lograda, en un intervalo de tasas de éxito, *recall* comprendido de i a j, a nivel gráfico representa el área de un rectángulo, dentro del intervalo de medición.

Y para obtener solo un indicador de *precisión*, que consideré todas las categorías de un algoritmo se calcula el promedio de las precisiones medias, identificada como **mAP**.

$$mAP = \sum_{c=1}^{\alpha} AP_c \tag{4.13}$$

AP y mAP son las dos medidas de referencia, que mas se utilizan para comparar evaluaciones sistemas de reconocimiento de objetos en imágenes.

4.3.4 Curva característica operativa del receptor, ROC

Otra herramienta gráfica aplicada para medir el desempeño del clasificador, es la **curva ROC**, formada por la memoria de cálculo de (4.11) y la tasa de *falsos positivos* [83], denominada en inglés como *fall-out* especificado para una clase en particular como

$$\mathcal{F}_c = \frac{FP_c}{TN_c + FP_c} \tag{4.14}$$

A pesar de no ser tan común como las *curvas recall-precisión*, también es utilizada para comparar el rendimiento de varios clasificadores, respecto a su asignación a un categoría c.

La utilidad de un método de clasificación es mesurada durante su proceso de *aprendizaje* y después de finalizarlo, aplicando diferentes indicadores. Al inicio se hace con la finalidad de ajustar los parámetros involucrados en el entrenamiento, pretendiendo obtener un modelo de categorización especifico y flexible, para lograr un desempeño aceptable al procesar observaciones que no había observado antes.

Resumen

El reconocimiento de objetos en imágenes, están fundamentados en algoritmos de *aprendizaje automático*, enfocados en particular para *clasificar* señales. Estos métodos son modelos matemáticos formados por criterios de distinción, cuyos parámetros se ajustan de forma especific, para cada aplicación, mediante un proceso iterativo denominado *aprendizaje*, guiado por un error de comparación que se define de acuerdo a las señales que se manejen.

Los algoritmos que han demostrado un buen funcionamiento, para analizar objetos en circunstancias poco controladas, generalmente están formados por el ensamble de *clasificadores* base, por esto en la Sección 4.1, se describieron los métodos que dan soporte a las formulaciones de análisis, que fueron consideradas mas aptas para este proyecto.

Dichas metodologías de *clasificación* y *detección* enfocadas en la detección de personas, se describieron en la Sección 4.2, y consisten de estrategias para analizar y describir objetos, derivadas del principio de *estructuras pictóricas* PS. Se presentó una síntesis de tres esquemas, en las cuales se resaltaron los conceptos que más podrían ayudar a cumplir con los objetivos planteados. Estos algoritmos son los modelos de partes deformables DPM, las redes neuronales convolucionales regionales R-CNN y algunas propuestas que combinan los primeros dos métodos.

Y por último en la Sección 4.3, se presentaron algunos de los indicadores estadísticos más comunes, para evaluar el desempeño de la toma de decisiones de un *clasificador* y comparar su rendimiento con otras metodologías utilizando *datasets* de referencia, como VOC. Con esto se concluye la revisión del marco teórico vinculado al sistema de detección, que se propusó y describe en el Capítulo 5.

Capítulo 5

Sistema de detección implementado

5.1 Descripción de funcionamiento

En la Figura 5.1 se muestra el diagrama del sistema de implementación propuesto, para detectar víctimas humanas no superficiales y parcialmente visibles entre escombros, de acuerdo con la metodología planteada en la Sección 1.4.

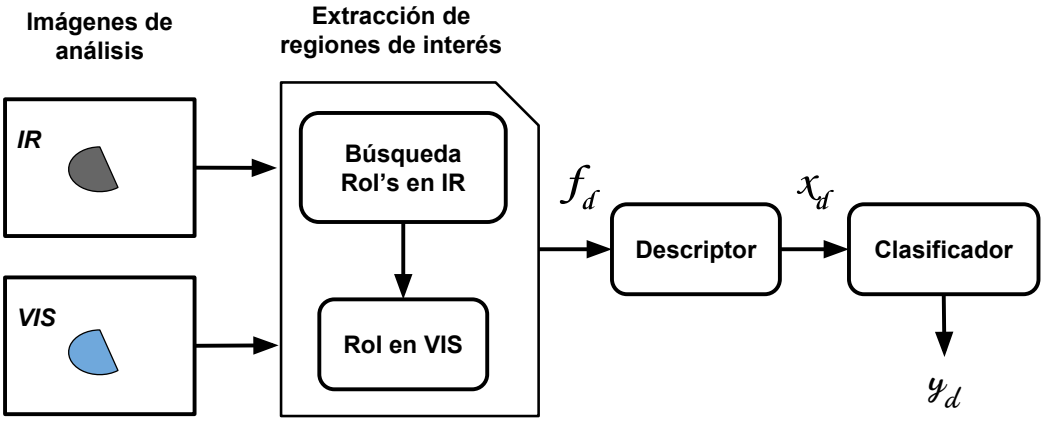


Figura 5.1. Estructura modular del sistema de detección de personas propuesto.

La primera etapa denominada *extracción de regiones de interés*, pretende afrontar en medida de lo posible, las condiciones y factores descritos en la Sección 1.2, para buscar y localizar zonas en la imagen, que puedan contener la proyección de una persona.

Al disponer de una cámara LWIR, se decidió utilizar como *referencia de búsqueda* la radiación de calor de los cuerpos, expresada en las imágenes IR como colecciones de pixeles que satisfacen un conjunto de criterios de selección, que se describirán posteriormente. de los cuales se determinan regiones de interés denominadas RoI, con un bounding box $\beta_{a,d} = [i_t, j_t, i_b, j_b]^T$ especificado por

su punto superior izquierdo (i_t, j_t) y el punto inferior derecho (i_b, j_b).

Del arreglo de coordenadas $\beta_{a,d}$, se localiza la RoI en la imagen VIS, aplicando la regla de correspondencia estimada, obtenida del montaje macroscópico estilo estereoscópico de las cámaras y del procedimiento de equivalencia de puntos, obteniendo el arreglo correspondiente del bounding box $v_{a,d} = [i_t, j_t, i_b, j_b]^T$.

Las otras dos etapas del sistema, implementan tres metodologías de *clasificación* inspiradas en los algoritmos presentados en la Sección 4.2. Dichos procedimientos se propusieron considerando; aprovechar la información de ambas cámaras, y experimentar con diferentes estrategias la capacidad de los *descriptores* de características.

5.2 Búsqueda y extracción de regiones

En esta Sección se describen los procedimientos desarrollados, para implementar obtener las señales de entrada del sistema, de acuerdo al marco teórico del Capítulo 2 y aspectos operativos establecidos en la Sección 1.3.

5.2.1 Adquisición de imágenes

Se utilizó la infraestructura de software propuesta e implementada en [1], la cual utiliza el *middleware* ROS para poder trabajar con las transmisiones de vídeo de ambas cámaras, realizar grabaciones, regular los cuadros por segundo, entre otros aspectos, que cubren el propósito de poder utilizar el sistema de detección, en un vehículo de exploración.

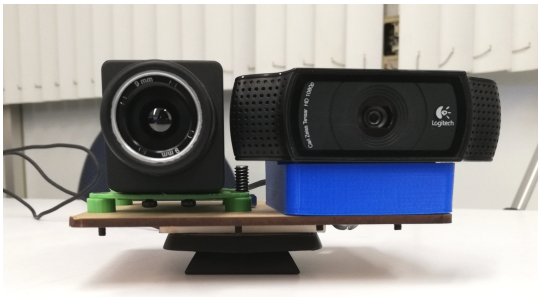
Este módulo de adquisición opera los siguientes modelos de equipos; webcam Logitech C920 y una cámara LWIR FLIR A35, mostradas en la Figura 5.2. Las especificaciones de ambos equipos están disponibles en [1].

(a) Cámara web Logitech C920. (b) Equipo FLIR A35, cámara térmica.

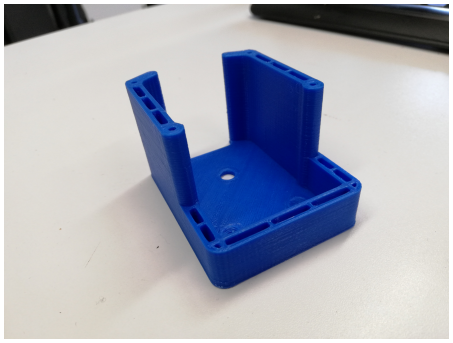
Figura 5.2. Cámaras del sistema de detección

Para lograr obtener un campo de visión compartido y establecer la correspondencia de puntos entre ambas cámaras, se diseñó una base para montar ambas cámaras, procurando alinear las lentes sobre el eje horizontal, a nivel macroscópico, como se observa en la Figura 5.3.

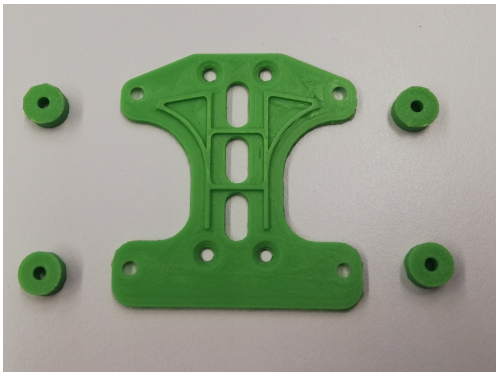
(a) Vista lateral del montaje de las cámaras.



(b) Vista frontal del montaje de las cámaras.



(c) Base de sujeción de la cámara web Logitech C920.



(d) Base de sujeción de la cámara LWIR FLIR A35.

Figura 5.3. Montaje estilo estereoscópico para alineación horizontal de las cámaras.

A cada una de las cámaras se le diseñó un soporte individual, para lograr alinearlas también en el eje vertical. Ambas bases fueron manufacturadas con una impresora 3D, y se sujetaron a un plano de MDF de tres milímetros de grosor.

5.2.2 Correspondencia de pixeles VIS-IR

Para obtener esta relación, como se mencionó en la Sección 2.2.2 se requiere un conjunto de puntos de referencia, identificables para ambas cámaras. En [57] utilizan un *patrón de calibración* tipo tablero de ajedrez, que reportan esta formado con papel aluminio y recortes de cuadros negros, obtenidos de una impresión láser en papel bond, como se observa en la Figura 5.4.

(a) Patrón de calibración.

(b) Observación VIS.

(c) Observación IR.

Figura 5.4. Imágenes del patrón de calibración utilizado en [57].

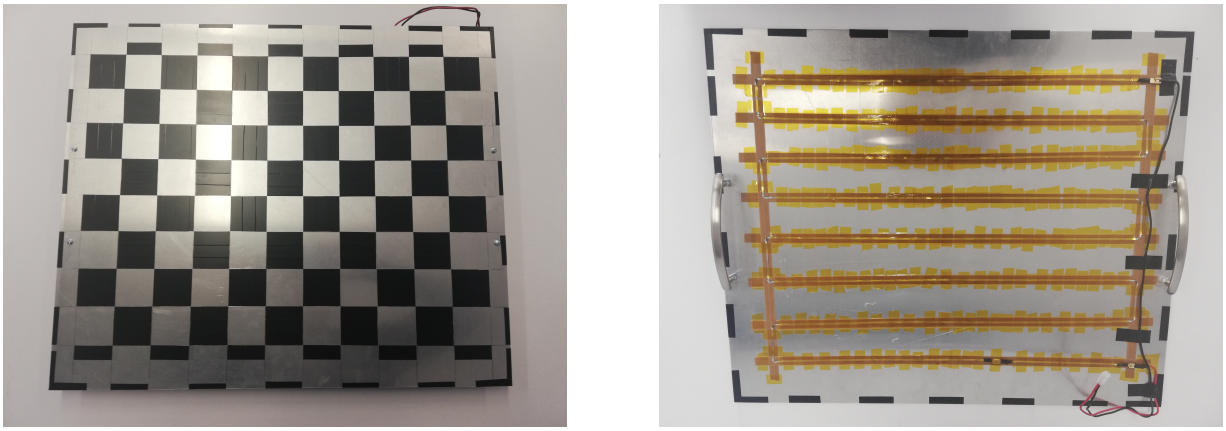
A pesar de no mencionar la razón de ocupar estos materiales, se consideró como punto de referencia para hacer el patrón de calibración. Las cámaras LWIR captan la proporción de irradiación infrarroja que emiten los cuerpos o superficies de objetos, denominada como ***emisividad***, que es proporcional a su temperatura promedio [1].

Esta capacidad de irradiación, atribuida principalmente por el material y acabado de la superficie de los objetos, es cuantificada por un coeficiente adimensional ε, que representa la fracción de irradiación en relación a un cuerpo negro, de $\varepsilon_0 = 1$. Utilizando el ***coeficiente de emisividad***, la ley de Stefan-Boltzman y otros aspectos es como las cámaras LWIR, logran estimar la temperatura de los objetos sin contacto alguno [89].

Ciertos materiales metálicos, al tener superficies pulidas u oxidadas, tienen *coeficientes de emisividad* del orden de centésimas por ejemplo, el papel aluminio tiene un coeficiente de $\varepsilon_{Al_f} = 0.03$ [89], por esta razón al observarlo junto con otro material de un coeficiente mucho mayor, se logra obtener una diferencia de contraste en una imagen infrarroja, como se observa en las Figuras 5.4.

La estructura de las casillas de un tablero de ajedrez, es un entramado que permite definir un conjunto de puntos de referencia, por el cambio de contraste que existe en los vértices de los cuadrados, por esto es utilizado con los colores blanco y negro para calibrar cámaras VIS. La visualización de estos puntos en la cámara IR, también es posible si se utilizan materiales con diferentes *coeficientes de emisividad*, para elaborar el patrón.

Entonces, tomando en cuenta estos elementos, se trazó un tablero de ajedrez sobre una **lámina de aluminio calibre 12**, con una superficie pulida. La mitad de la estructura de los cuadrados, se hizo con **cinta de vinil color negro** para aislamiento eléctrico, porque tiene un *coeficiente de emisividad* de $\varepsilon_v = 0.98$. De manera que, el patrón de calibración propuesto se muestra en la Figura 5.5a.



(a) Tablero de ajedrez trazado. (b) Reverso del tablero de ajedrez.

Figura 5.5. Fotografías del patrón de calibración construido para el proyecto.

A diferencia de [56–59], se decidió calentar el patrón de calibración, utilizando una resistencia formada por 2.5 metros de **cinta nicromel** de 5.8 $[\Omega/m]$, la cual se adhirió en la otra cara de la lámina, como se muestra en la Figura 5.5b. Esta modificación pretende tener un contraste constante, que no se pierda al enfriarse la lámina, o depender de las condiciones ambientales, como se observa en la Figura 5.4c.

Los cuadrados miden cinco centímetros por lado, y la estructura central esta formada de 9×8 casillas como se muestra en la Figura 5.5a.

5.2.3 Búsqueda de la región de interés

La cámara FLIR A35 cuenta con la opción de regular el rango dinámico de su escala de grises, de acuerdo a un intervalo absoluto o relativo de temperatura o ajustándose automáticamente de acuerdo a la radiación de los objetos que tenga en escena, estableciendo su punto máximo en la fuente de mayor calor.

De estas configuraciones, se decidió utilizar el *ajuste automático* de la escala de grises, porque se observó que de esta manera, se resaltan más las siluetas de los cuerpos que irradian calor, cuya temperatura fuera superior a la del medio ambiente, hecho que favorece el análisis descriptivo.

Entonces, la localización de regiones de imagen para analizar, se realizó implementando un *filtro de umbral* de pixeles, para determinar un bounding box considerando que las regiones seleccionadas, tuvieran un área mínima, de acuerdo al estadístico obtenido, al realizar las anotaciones de es del *dataset*.

5.3 Metodologías de clasificación

En los Capítulos previos, se expusieron diferentes estrategias y procedimientos de sistemas de detección y clasificación, que fueron revisados por considerar circunstancias semejantes, a las especificadas en la Sección 1.2.

Como se mencionó en el Capítulo 4, analizar y describir un objeto modelándolo como la composición de diversas partes, es una robusta estrategia implementada en la actualidad, en diferentes sistemas de detección reconocidos por lograr altos índices mAP, al evaluarse con *datasets* canon, como VOC [23].

A pesar de que estas pruebas se hacen con imágenes de cámaras VIS, en diferentes aplicaciones se ha experimentado con el desempeño de dichos algoritmos, en sistemas como el planteado en la Figura 5.1 e incluso con arreglos de cámaras y otros sensores, logrando resultados favorables para sus fines [33–36, 56].

De acuerdo los posibles estados, en los que se pueda encontrar una víctima no superficial, el esquema de *aprendizaje* y descripción por regiones, planteado en R-CNN es el que se decidió

utilizar como guía, para proponer experimentar con los siguientes métodos de clasificación, de acuerdo a las imágenes de trabajo disponibles.

5.3.1 CNN VIS AlexNet/VGG16

Esta propuesta consiste en utilizar transfer learning, de la arquitecturas AlexNet y VGG16 mostradas en las Figuras 3.4 y 3.5, para aplicarle un *ajuste fino* a las penúltimas capas y determinar los parámetros de la última capa, para que funcioné como ***clasificador binario***, utilizando la *función de activación* (3.28c).

Con estos dos *clasificadores* se pretendió experimentar con la capacidad descriptiva, de estas dos arquitecturas a partir de los conjuntos $\mathcal{W}$ y $\mathcal{B}$ que se definieron al entrenar las redes con el *dataset* ImageNet. La prueba consistirá en incorporar en la etapa de *reajuste* de ciertos parámetros de las últimas capas, imágenes IR, y ver el desempeño que puede lograrse con esta variación.

5.3.2 CNN's IR-VIS

Inspirado en la arquitectura y señal de entrada utilizada en [27], con la finalidad de hacer un análisis que incluya a los dos tipos de imagen, se propone probar un *clasificador* estructurado en una red neuronal tipo convolucional, que tenga como señales de entrada, dos mapas de características extraídos de las imágenes IR y VIS. La arquitectura propuesta es la mostrada en la Figura 5.6

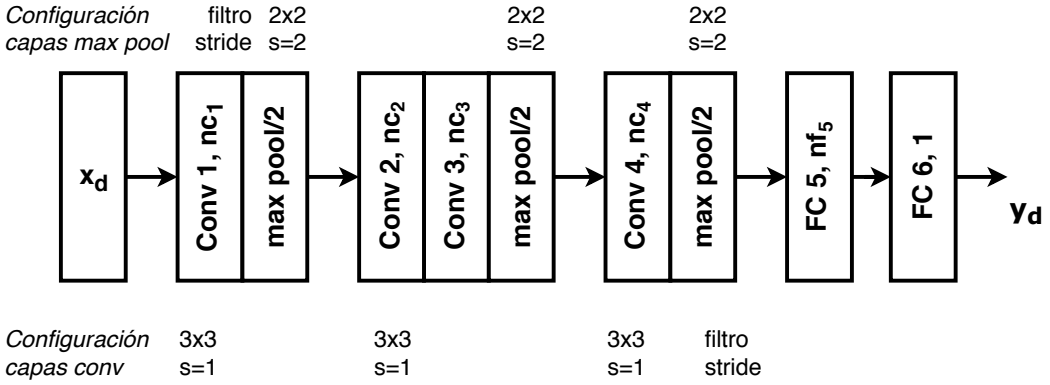


Figura 5.6. Arquitectura de red neuronal para procesar mapas de características x_d.

El diseño de la estructura, se propuso considerando la composición de AlexNet, VGG16 y el planteamiento de Ouyang, mostrado en la Figura 4.6. El número de neuronas en cada capa, se especifica por una variable porque, se ajustaron para entrenar tres modelos que utilizaron como entrada, mapas o conjuntos de características HOG, SIFT y SURF.

De manera que para cada *clasificador*, x_d **esta compuesta por dos arreglos 2D, uno por cada imagen IR y VIS**, aplicando los descriptores expuestos en la Sección 3.3, como se especifica a continuación.

Tensor de entrada HOG

Durante la documentación del descriptor HOG se encontró que se pueden formar imágenes de bordes, a partir de estos histogramas, en los cuales se resaltan más las siluetas de los objetos contenidos en las proyecciones, comparados con los resultados de aplicar otros algoritmos. Por ello, el primer tensor x que se propone para probar, la red CNN IR-VIS es el siguiente

$$x_d = \{\mathcal{H}_{\mathrm{I}_d}, \mathcal{H}_{v_d}\}_{d=1}^{N} \tag{5.1}$$

en donde los subíndices I y v son utilizados para diferenciar la fuente de extracción de los mapas de bordes, es decir de las imágenes IR o VIS, respectivamente. Las dimensiones de x son de 224×224×2, y en caso de que la RoI seleccionada sea mayor, se ajusta a dicho tamaño. El conjunto de histogramas para formar $\mathcal{H}_{\mathrm{I}||v}$, se calculan con el Algoritmo 3.3.1, modificando únicamente el tamaño de la imagen de entrada.

Tensor de entrada SIFT

Esta señal de entrada esta formada por dos conjuntos de 90 puntos característicos SIFT, extraídos de cada una de las imágenes IR y VIS, aplicando el Algoritmo 3.3.2. El número de puntos característicos utilizados en este tensor, se determinaron de acuerdo a la media estadística, que se obtuvo del *dataset* empleado para el entrenamiento de la arquitectura CNN IR-VIS, y que se describe más adelante.

Cada uno de los arreglos 2D de tamaño 90×128, que integran el tensor x_d de este clasificador, es una lista de 150 puntos seleccionados de forma aleatoria, de todos los que se obtienen en una RoI. En cada arreglo, se acomodaron por renglones de acuerdo a su posición (i_k, j_k), que es parte de los elementos que componen el vector característico

$$pc_k = [i_k, j_k, \sigma_k, \theta_k, vd_k] \tag{5.2}$$

de manera que el tensor de entrada para esta versión del clasificador, es el siguiente

$$x_d = \{\xi_{\mathrm{I}_d}, \xi_{v_d}\}_{d=1}^{N} \mid \xi_{\mathrm{I}/v_d} = [pc_1, pc_2, \dots, pc_{150}]^T \tag{5.3}$$

Tensor de entrada SURF

El procedimiento de formación de esta señal de entrada, es prácticamente el mismo explicado para (5.3), exceptuando que el *descriptor* aplicado es el del Algoritmo 3.3.3, para extraer de cada imagen IR y VIS, un conjunto de *puntos característicos*, en donde cada uno esta definido por la siguiente configuración

$$pc_k = pc_k = \left[i_k, j_k, L_k, \theta_k, vd_k, \mathcal{L}_{sign_k}\right] \tag{5.4}$$

y en consecuencia, el tensor de entrada definido al utilizar el descriptor SURF es

$$x_d = \{\pi_{\mathrm{I}_d}, \pi_{v_d}\}_{d=1}^{N} \mid \pi_{\mathrm{I}/v_d} = [pc_1, pc_2, \dots, pc_{150}]^T \tag{5.5}$$

5.3.3 Colección de imágenes de trabajo

Para llevar a cabo los entrenamientos y/o ajustes finos de los *clasificadores* propuestos para el sistema de detección, se hizo una recopilación de imágenes de diferentes *datasets*. Como se mencionó en las Secciones 3.3.4 y 4.1, para entrenar un algoritmo de *aprendizaje automático* se necesitan muestras representativas, de los objetos que se desean identificar, en cantidades suficientes para obtener patrones estadísticos.

El inconveniente con el desarrollo de la aplicación propuesta, es la escasez de imágenes de personas en situaciones post-desastre reales o semejantes, por eso la mayoría de los trabajos revisados en los antecedentes, crean sus propias colecciones de datos, acción que también se llevó a cabo, sobretodo por utilizar cámaras de diferente sensibilidad.

Por esta razón, se propuso experimentar con los tres *clasificadores* CNN IR-VIS, mencionados anteriormente. Para ello, se realizó una recopilación de imágenes de diferentes *datasets*, especificados en la Tabla 5.1, que se denomino UNAM IRVIS-ext, que se denotará como $\Lambda = \{\mathrm{I}_j\} \mid j \epsilon \mathbb{N}; j = 1, \ldots, N_\Lambda$, donde N_Λ es el número total de imágenes de la colección.

Tabla 5.1. Registro de imágenes tomadas de cada dataset, para formar UNAM IRVIS-ext.

Dataset	Imágenes VIS	Imágenes IR	*Clase 1*	*Clase -1*
INRIA [18]	400	-	200	200
VOC 2012 [23]	500	-	250	250
IDV-50 [21]	50	-	50	-
Kaist [90]	1300	1300	1300	1300
UNAM IRVIS	1200	1200	1200	1200

La *Clase 1*, es la etiqueta asignada a las imágenes que contienen a una persona, y el caso opuesto es identificado como la *Clase -1*. La mayoría de las imágenes son de peatones, capturados en diferentes entornos, distancias e incluyendo en algunos casos oclusiones. La distribución del total de muestras tomadas de cada *dataset*, fueron de 50 % para cada clase.

Al utilizar estas imágenes se pretende formar una base general descriptiva, de distintas observaciones del cuerpo completo de las personas, por ello también se consideraron imágenes de VOC, que contienen personas realizando diferentes actividades, para considerar variaciones.

El *dataset* Kaist [90], es el único que se encontró con pares de imágenes IR y VIS, semejantes a las utilizadas en el proyecto. Por lo que, se utilizó como complemento de la colección propuesta para este trabajo, denominada UNAM IRVIS que contiene imágenes de personas, debajo de escombros y de observaciones de partes o cuerpos completos, que se estiman pueden presentarse en una situación real.

Del conjunto Λ, se formaron dos subconjuntos para entrenar y evaluar los modelos:

- Λ_v esta formado únicamente por las 3350 imágenes VIS, para ser utilizado por los clasificadores CNN VIS AlexNet/VGG16.
- Λ_{iv} contiene 2500 pares de muestras, provenientes de los *datasets* Kaist y UNAM IRVIS, para ocuparse de los experimentos con CNN IR-VIS.

En la siguiente Sección, se describirá como se organizaron las colecciones de imágenes, para efectos del proceso de aprendizaje y evaluación del modelo.

5.3.4 Entrenamiento de los clasificadores

Para realizar el proceso de aprendizaje de los *clasificadores* propuestos, los dos subconjuntos del *dataset* Λ, se organizaron en las tres divisiones necesarias para el proceso de aprendizaje y evaluación del modelo, manteniendo la regla de fraccionamiento de 80/10/10 %. Las distribuciones del conjunto Λ_v, se denotarán como

$$\mathrm{T}_v = \left\{\left(\nu_j, \mathcal{Y}_j, \beta_j\right)\right\}_{j=1}^{N_{\mathrm{T}_v}} \quad (5.6a) \qquad \mathcal{V}_v = \left\{\left(\nu_j, \mathcal{Y}_j, \beta_j\right)\right\}_{j=1}^{N_{\mathcal{V}_v}} \quad (5.6b) \qquad \mathrm{E}_v = \left\{\left(\nu_j, \mathcal{Y}_j, \beta_j\right)\right\}_{j=1}^{N_{\mathrm{E}_v}} \quad (5.6c)$$

en donde $\mathcal{Y}_j \epsilon\, [1, -1]$ es la clase de la muestra y β_j, corresponde al bounding box registrado como real para cada imagen. Respecto al conjunto Λ_{iv}, la notación es la siguiente

$$\mathrm{T}_{iv} = \left\{\left(\mathrm{I}_j, \nu_j, \mathcal{Y}_j, \beta_j\right)\right\}_{j=1}^{N_{\mathrm{T}_{iv}}} \quad (5.7a) \qquad \mathcal{V}_{iv} = \left\{\left(\mathrm{I}_j, \nu_j, \mathcal{Y}_j, \beta_j\right)\right\}_{j=1}^{N_{\mathcal{V}_{iv}}} \quad (5.7b) \qquad \mathrm{E}_{iv} = \left\{\left(\mathrm{I}_j, \nu_j, \mathcal{Y}_j, \beta_j\right)\right\}_{j=1}^{N_{\mathrm{E}_{iv}}} \quad (5.7c)$$

Para realizar el entrenamiento de los *clasificadores*, se implementó el esquema de regiones propuestas del algoritmo R-CNN, descrito en la Sección 4.2.2. Para llevar a cabo este procedimiento, se formaron *mini batches* de 16 regiones por cada muestra de entrenamiento, utilizando como *función de costo* la *cross entropia binaria* (3.32b) y el optimizador RMSProp (3.36). Se aplicó esta metodología, para que probar si las redes, pueden *aprender* a describir una persona, mediante observaciones parciales.

Parámetros de entrenamiento de CNN VIS

Los modelos CNN VIS AlexNet y VGG16, se entrenaron a partir del *transfer learning* hecho de la arquitecturas, entrenadas con el dataset ImageNet, utilizando el conjunto de parámetros de estas estructuras, disponibles en la biblioteca[1] de *aprendizaje profundo*, Keras.

De las arquitecturas originales mostradas en las Figuras 3.4 y 3.5, se sustituyeron las últimas capas de cada una, por otra de una neurona con la función de activación (3.28c). Y los parámetros $\mathcal{W}$ y $\mathcal{B}$ de las dos penúltimas capas, se reiniciaron junto con la última, con los métodos de inicialización de [28].

[1]En referencia a un compendio de métodos y herramientas de software, en este caso para el lenguaje de programación Python

Para realizar el *ajuste fino* de las arquitecturas, como se mencionó en la Sección 5.3.1 se declararon como constantes los $\mathcal{W}$ y $\mathcal{B}$, de las siguientes capas:

- *conv1* a *conv4* para AlexNet
- *conv1-1* a *conv3-2* para VGG16

Y el resto de los *pesos sinápticos* y coeficientes de ajuste, se ajustaron con otro proceso de entrenamiento, utilizando solo los *dataset* (5.6a) y (5.6b). En la Tabla 5.2, se muestra los valores iniciales de los hiper parámetro involucrados en el entrenamiento, para cada caso, recordando que se uso como optimizador RMSProp (3.36).

Tabla 5.2. Valores iniciales de los parámetros de entrenamiento de CNN VIS.

Arquitectura	Epochs	Ponderador β	Tasa de aprendizaje γ	Muestras por batch	ε
AlexNet	180	0.01	1E-05	320	10E-08
VGG16	150	0.015	2E-05	250	10E-08

Estos valores, fueron los correspondientes de cada modelo entrenado, que obtuvo el mejor rendimiento durante su evaluación, que se mostrará en el siguiente Capítulo.

Parámetros de entrenamiento de CNN IR-VIS

Para las tres versiones de este *clasificador*, descritas en la Sección 5.2.2, el *aprendizaje* comenzó desde cero, porque no existen antecedentes de arquitecturas, públicas para su utilización que usen la combinación de imágenes IR-VIS. La arquitectura propuesta CNN IR-VIS, fue resultado de experimentar, con pocos *epochs* el rendimiento de otros modelos, para llegar a seleccionar la de la Figura 5.6.

En la Tabla 5.3 además de los *hiper parámetro* utilizados para el entrenamiento, también se especifican el número de neuronas por capa, de aquellas indicadas como variables en la Figura 5.6, de cada una de las versiones.

Tabla 5.3. Valores iniciales de los parámetros de entrenamiento para CNN's IR-VIS.

Arquitectura	Epochs	Ponderador β	Tasa de aprendizaje γ	Muestras por batch	ε	nc_1	$nc_2 = nc_3$	nc_4	nf_5
CNN IR-VIS HOG	280	0.01	1E-05	600	10E-08	32	64	64	256
CNN IR-VIS SIFT	250	0.008	1E-04	600	10E-08	32	64	64	512
CNN IR-VIS SURF	300	0.005	2E-04	600	10E-08	32	64	64	512

5.3.5 Software y hardware de implementación

Los *clasificadores* propuestos, se implementaron en el lenguaje Python 3.0 en conjunto con el uso de las siguientes bibliotecas de métodos y herramientas de procesamiento de imágenes y/o dedicadas a modelos de aprendizaje automático; Keras, TensorFlow, OpenCV, scikit-learn y scikit-image. Dicho conjunto de instrumentos se instalaron en el sistema operativo Ubuntu 16.04 LTS, junto con el mencionado *middleware* ROS en su versión Kinetic.

Para entrenar los modelos, se utilizó una tarjeta gráfica NVIDIA GeForce GTX 1060, de 6 GB de memoria RAM dedicada, conectada a un procesador Intel Core i7 de cuarta generación, con 8 GB de memoria RAM.

Resumen

En este Capítulo se presentaron las actividades y procesos desarrollados, para la implementación del sistema de detección de personas en escenarios post-desastre, el cual fue orientado a víctimas no superficiales atrapadas bajo escombros. En la Sección 5.1 se describió el funcionamiento general, del sistema propuesto para que pueda funcionar en línea, después de obtener un modelo de clasificación funcional.

Posteriormente, en la Sección 5.2 se presentaron las cámaras que se utilizaron para crear el *dataset* de imágenes UNAM IRVIS, así como la manera en la que se montaron en una base, para que tuvieran un *campo de visión* compartido. A partir de observar la misma escena, se especificó la construcción del *patrón de calibración* ideado, para poder establecer la correspondencia de puntos a nivel pixel.

De los algoritmos revisados en el Capítulo 4, se seleccionó ajustar el planteamiento de R-CNN, para entrenar y formular los cuatro clasificadores que se propusieron. Estos algoritmos fueron descritos en la Sección 5.3, en conjunto con algunas de sus especificaciones de implementación.

Capítulo 6

Pruebas y resultados del sistema

En este Capítulo se expone el registro de resultados obtenidos de cada experimento realizado, para evaluar el funcionamiento y desempeño del sistema propuesto para detectar personas descrito en el Capítulo 5.

Cada una de las pruebas realizadas se describe en las diferentes Secciones, con el objetivo de establecer un contexto que favorezca la interpretación del o los resultados reportados y poder concluir la operatividad de cada bloque que integra el esquema de localización de víctimas planteado.

De acuerdo al flujo de procesamiento indicado en la Figura 5.1, primero en la Sección 6.1 se presentan los parámetros estimados del *modelo de proyección* (2.17) de cada una de las cámaras empleadas y los errores de reproyección obtenidos. Después, las matrices intrínsecas y coeficientes de distorsión óptica calculados, se ocuparon para buscar la *relación de correspondencia* de puntos entre las imágenes IR y VIS a nivel pixel, aplicando dos de los procedimientos descritos en el Capítulo 2.

Y en la Sección 5.3 se reporta la formulación de evaluación estadística que se aplicó a cada *clasificador*, con el propósito de mensurar su capacidad de catalogación generada por el proceso de entrenamiento, cuyo progreso se registró en las respectivas *curvas de aprendizaje* y pruebas con imágenes que no participaron en la estimación de parámetros de las redes CNN.

6.1 Calibración y correspondencia de pixeles

De acuerdo con el desarrollo teórico expuesto en el Capítulo 2, para obtener la relación de correspondencia de *pixeles* entre las imágenes IR y VIS, primero se necesitan estimar los parámetros que definen el *modelo de proyección especifica* (2.17), de cada una de las cámaras tomando en cuenta que ambas pueden observar un mismo conjunto de puntos de referencia, dentro de su *campo de visión* compartido.

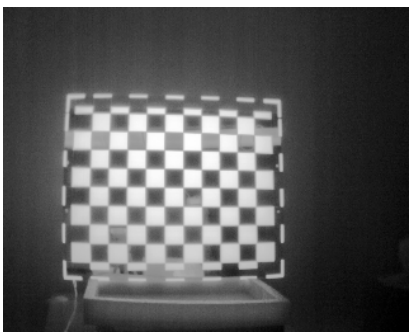
Por esta razón, la primera división de esta Sección describe el procedimiento aplicado para *calibrar* cada cámara monocular y presenta los resultados obtenidos. En el segundo apartado, se expone el desarrollo de los métodos de correspondencia de puntos entre ambos tipos de imágenes, utilizando los coeficientes de distorsión y parámetros intrínsecos de cada cámara.

6.1.1 Calibración de las cámaras monoculares

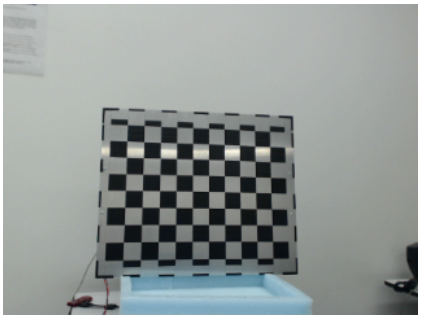
La cámara FLIR A35 provee proyecciones de 320×240 pixeles en escala de grises, mientras que la webcam Logitech C920 también tiene dicho tamaño pero como resolución mínima, disponiendo de imágenes haste de 1920×1080. Por esta diferencia de tamaños y tomando en cuenta las proporciones del tensor de entrada de las arquitecturas AlexNet y VGG16, *se determinó utilizar la resolución espacial* de 640×480 pixeles para ambas cámaras, que es igual a las muestras del dataset KAIST [90].

Para revisar cuanto influye la duplicación de tamaño de las imágenes LWIR, para la correspondencia de puntos entre las imágenes, se decidió observar la diferencia entre los *errores de reproyección* generados al aplicar el método de calibración usando imágenes en su resolución fuente y al tamaño de trabajo citado en el párrafo anterior.

De manera que para cada cámara se aplicó el Algoritmo 2.2.2, utilizando el patrón de calibración descrito en la Sección 5.2.2 y mostrado en la Figura 6.1. La implementación del método mencionado se hizo con apoyo del paquete *camera_calibration*, disponible como una de las herramientas de ROS.



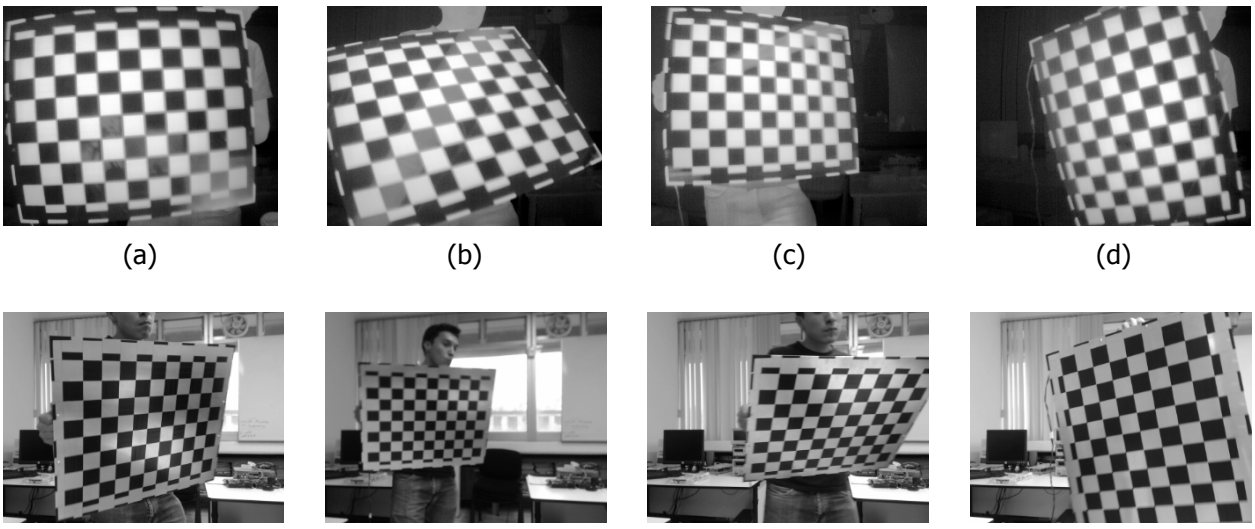
(a) Proyección en cámara IR.



(b) Proyección en cámara VIS.

Figura 6.1. Observación del patrón de calibración, dentro del campo de visión compartido.

Los parámetros intrínsecos (2.16) y coeficientes de distorsión óptica (2.14) de cada cámara, se seleccionaron de siete conjuntos estimados en diferentes iteraciones, utilizando en cada experimento un promedio de 100 imágenes del *patrón de calibración*, colocado en diferentes posiciones en un intervalo de distancia entre el plano y las cámaras de 0.3 a 1.5 metros, como se muestra en algunas de las imágenes capturadas en el procedimiento, en la Figura 6.2.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figura 6.2. Imágenes de muestra, de los conjuntos que fueron utilizados para el proceso de calibración, en las dos resoluciones.

En cada imagen se detectaba una malla de 8×7 puntos, ubicados en los vértices del entramado de cuadros de 5×5 centímetros, distinguibles por el cambio de contraste. De los siete grupos de parámetros obtenidos se seleccionó el conjunto más cercano a la media de la colección, como los factores característicos de cada cámara, registrando en la Tablas 6.1 y 6.2 los parámetros intrínsecos y coeficientes de distorsión respectivamente.

Cámara	Resolución	α	β	c_x	c_y
FLIR A35	320×240	366.0902	365.1015	166.0042	123.9225
	640×480	729.2916	681.9432	320.2283	243.7117
Logitech C920	320×240	304.6982	307.327	162.79	124.25
	640×480	633.4308	633.421	314.162	235.0238

Tabla 6.1. Parámetros intrínsecos de cada una de las cámaras.

Cámara	Resolución	k_1	k_2	p_1	p_2
FLIR A35	320×240	-0.4485	0.1888	-0.0012	0.0005
	640×480	-0.42901	0.03005	0.00148	-0.00101
Logitech C920	320×240	0.1382	-0.2535	-0.0021	0.0062
	640×480	0.1285	-0.2222	-0.0075	0.0009

Tabla 6.2. Coeficientes de distorsión óptica, de cada una de las cámaras.

Utilizando estos parámetros y la colección de imágenes que los permitió estimar, se calcularon los *errores cuadráticos de reproyección* (2.30a) promedio, para cada una de las resoluciones registrando los resultados en las Tablas 6.3a y 6.3b, en donde las diferencias de proyección están medidas en pixeles.

Cámara	Resolución	E_{rp}
FLIR A35	320×240	0.1815
	640×480	0.2381

(a)

Cámara	Resolución	E_{rp}
Logitech C920	320×240	0.034
	640×480	0.0531

(b)

Tabla 6.3. Registro de errores de reproyección de cada cámara, en sus dos resoluciones.

Considerando la magnitud de los errores de reproyección, y la rectificación de las imágenes se puede considerar que los parámetros de calibración de las cámaras, son funcionales para la aplicación del proyecto.

6.1.2 Correspondencia de pixeles entre imágenes IR-VIS

Para determinar la correlación de pixeles entre las imágenes de ambas cámaras, se eligió experimentar con los métodos de *geometría epipolar* e *interpolación geométrica*, descritos en el Capítulo 2. En las siguientes Secciones se explica el desarrollo de cada uno de estos procedimientos y conjuntamente se presentan los resultados obtenidos.

Asociación por geometría epipolar

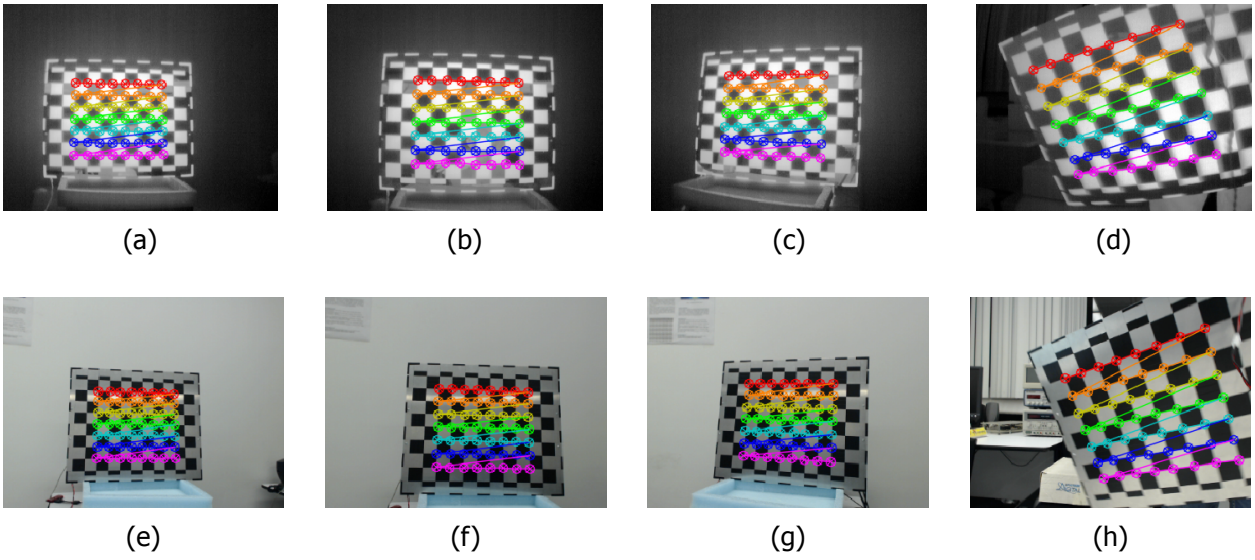
Para determinar la relación (2.39), se utilizaron los parámetros intrínsecos indicados en las Tablas 6.1 y 6.2 distribuidos en dos archivos *.yaml* para diferenciar las cámaras. Este formato se seleccionó principalmente por ser comúnmente utilizado en diversas aplicaciones de ROS y además es compatible con los lenguajes de programación C/C++ y Python.

Tomando como referencia la configuración de cámaras mostrada en la Figura 2.7 y la disposición física de su montaje presentado en la Figura 5.3b, el método fue desarrollado considerando como

cámara izquierda la C920 y como derecha la A35.

Con la finalidad de definir $\mathbf{R_E}$ y $\mathbf{t_E}$ de (2.39), se sincronizaron las transmisiones de vídeo para capturar las imágenes del *patrón de calibración*, en las cuales se detectó una malla de 8×7 puntos de referencia utilizando nuevamente el paquete *camera_calibration*, esta vez configurado en modo estereoscópico.

Utilizando un promedio de 60 imágenes de diferentes posiciones del tablero de ajedrez, mostrando algunas de las observaciones registradas en la Figura 6.4, se realizaron siete iteraciones del procedimiento obteniendo el mismo número de matrices esenciales, fundamentales y otros parámetros involucrados en la relación de correspondencia para cada una de las dos resoluciones.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figura 6.3. Imágenes de muestra de los conjuntos utilizados para estimar la correspondencia de puntos, por medio de *geometría epipolar*.

De la colección de parámetros estimados de cada experimento, se seleccionó un solo conjunto aplicando el mismo criterio de elegir el más cercano a la media, como se mencionó en la Sección 6.1.1, de tal manera que las matrices fundamentales y esenciales utilizadas para establecer la relación de correspondencia, para la resolución de 320×240 son las siguientes

$$\mathbf{F}_{320} = \begin{bmatrix} 3.0497 \times 10^{-6} & 4.3556 \times 10^{-5} & 0.00125 \\ -3.779 \times 10^{-5} & 3.9603 \times 10^{-6} & 0.0188 \\ -0.0047 & -0.0209 & 1.0 \end{bmatrix} \tag{6.1}$$

de la cual acorde a la expresión (2.38) y los parámetros intrínsecos registrados en la Tabla 6.1, la *matriz esencial* correspondiente es

$$\mathbf{E}_{320} = \begin{bmatrix} 3.1837 & 45.7483 & 23.7880 \\ -38.4408 & 4.0530 & 45.3058 \\ -27.7661 & -40.32837 & 6.8745 \end{bmatrix} \tag{6.2}$$

Y finalmente al descomponer la matriz (6.2) aplicando el método SVD, se obtuvieron las variables que determinaron la correspondencia (2.39) para esta resolución, las cuales son

$$\mathbf{t}_{\mathbf{E}_{320}} = \begin{bmatrix} -40.7580 \\ 27.9888 \\ -43.4227 \end{bmatrix} \quad (6.3) \qquad \mathbf{R}_{\mathbf{E}_{320}} = \begin{bmatrix} 0.9933 & -0.0094 & -0.1147 \\ -0.0008 & 0.9959 & -0.0898 \\ 0.1151 & 0.0893 & 0.9893 \end{bmatrix} \quad (6.4)$$

Este procedimiento también se desarrolló con los parámetros intrínsecos correspondientes a imágenes de 640×480, obteniendo los siguientes resultados

$$\mathbf{F}_{640} = \begin{bmatrix} -8.6588 \times 10^{-8} & -2.3096 \times 10^{-5} & 0.0229 \\ 2.1651 \times 10^{-5} & 1.9069 \times 10^{-6} & -0.0180 \\ -0.0251 & 0.0180 & 1.0 \end{bmatrix} \quad (6.5)$$

$$\mathbf{E}_{640} = \begin{bmatrix} 0.4144 & 110.4002 & -133.94951 \\ -100.9189 & -8.8765 & 81.49173 \\ 144.2354 & -80.4775 & -3.6779 \end{bmatrix} \quad (6.6)$$

$$\mathbf{t}_{\mathbf{E}_{640}} = \begin{bmatrix} -83.8674 \\ -142.3385 \\ -99.3508 \end{bmatrix} \quad (6.7) \qquad \mathbf{R}_{\mathbf{E}_{640}} = \begin{bmatrix} 0.9995 & 0.0217 & 0.0210 \\ -0.0233 & 0.9965 & 0.0795 \\ -0.0192 & -0.08003 & 0.9966 \end{bmatrix} \quad (6.8)$$

Con estas variables se calculó el *error cuadrático de reproyección promedio* para cada resolución, considerando las diferencias de proyección de ambas imágenes como [39]

$$\mathrm{E}_{rp} = \frac{1}{N}\left((2.29)|_{\nu} + (2.29)|_{\mathrm{I}}\right) \quad (6.9)$$

En esta expresión se suman los errores de reproyección (2.29) calculados para cada conjunto de puntos de referencia, detectados en las imágenes IR y VIS denotadas como I y ν respectivamente y N es la cantidad de observaciones del patrón de calibración. Los errores (6.9) obtenidos de la correspondencia establecida están medidos en pixeles y son los siguientes

Resolución	320×240	640×480
E_{rp}	0.6013	0.7613

Tabla 6.4. Errores de reproyección para correspondencia de pixeles usando el método de geometría epipolar

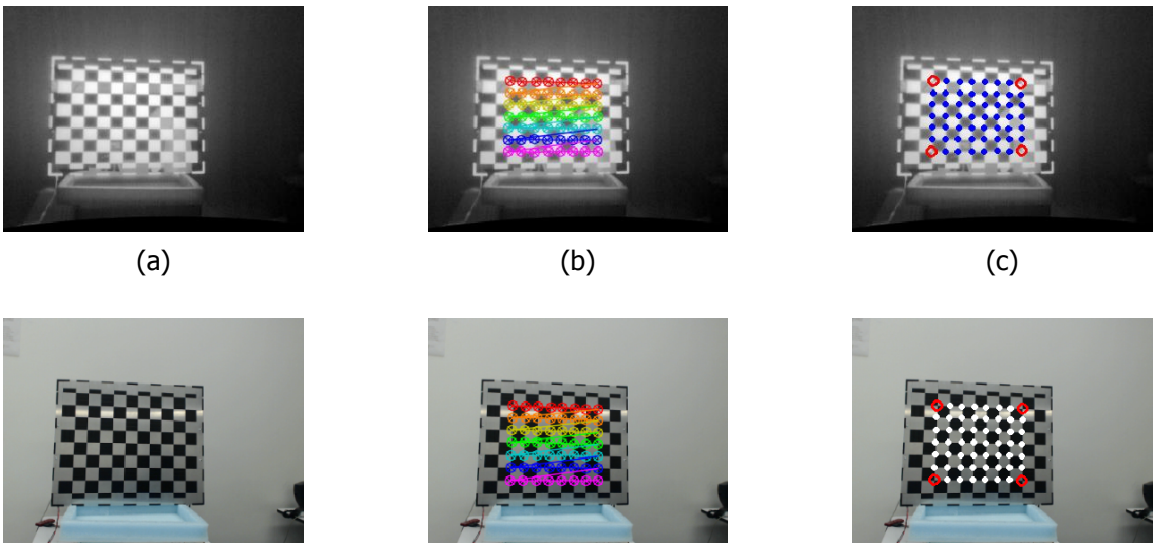
Interpolación geométrica

Para implementar este método explicado en la Sección 2.4.1, se utilizaron imágenes del patrón de calibración de ambas cámaras aplicando a cada una la corrección de distorsión óptica, utilizando los parámetros de las Tablas 6.1 y 6.2. Esta acción se argumenta porque las imágenes mostradas

en [56], aparentan tener la corrección aunque en la descripción del planteamiento no se especifica.

En primera instancia, la malla de 8×7 puntos de referencia fue determinada en ambas imágenes por la función empleada para detectar las esquinas del entramado del tablero de ajedrez, aprovechando que ya se contaba con este recurso y evitando establecerlos manualmente como se indica en [56].

Definidas las colecciones de marcas en las dos escenas del patrón de calibración, se seleccionaron cuatro puntos de cada imagen, señalados en las Figuras 6.4c y 6.4f con un círculo alrededor de cada punto, con el objetivo de plantear el sistema de ecuaciones (2.40) y resolverlo para obtener la *matriz de transformación geométrica* que define la correspondencia (2.41).



(a) (b) (c)

(d) (e) (f)

Figura 6.4. Detección de puntos de referencia para estimar la matriz de correspondencia.

Este planteamiento también se aplicó para las dos resoluciones de imágenes, obteniendo las matrices que establecen la relación de pixeles de la cámara IR → VIS, que se presentan a continuación

$$\mathbf{M}_{320} = \begin{bmatrix} 0{,}98501 & -0{,}0012 & -8{,}01842 \times 10^{-5} & 5{,}2584 \\ 0{,}00524 & 1{,}05803 & 2{,}2997 \times 10^{-5} & 8{,}67257 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{6.10}$$

$$\mathbf{M}_{640} = \begin{bmatrix} 0{,}9972 & 0{,}0038 & -4{,}4894 \times 10^{-5} & 13{,}8616 \\ 0{,}01523 & 1{,}0464 & -1{,}9933 \times 10^{-5} & 12{,}1466 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{6.11}$$

Posteriormente para comparar la diferencia de proyección que definen (6.10) y (6.11), utilizando la relación (2.41) se cálculo el *error de reproyección*. Para esto se consideraron como *referencia* las posiciones de los puntos restantes, que fueron detectados en las imágenes VIS, señalados con círculos sólidos como se muestran en la Figura 6.4f. Después la otra colección de puntos restantes

detectados en la imagen IR, fueron transformados al espacio de la proyección VIS para definir las estimaciones de correspondencia, utilizadas en (2.29) obteniendo las siguientes imprecisiones

Resolución	320×240	640×480
E_{rp}	0.3157	0.42814

Tabla 6.5. Errores de reproyección en pixel de la correspondencia IR → VIS, estimada por el método de interpolación geométrica.

6.2 Evaluación de clasificadores

El contenido de este apartado se compone principalmente de los registros de cálculo de los índices expuestos en la Sección 5.3, y algunas de las memorias de los indicadores utilizados para guiar y observar el proceso de aprendizaje de cada clasificador, especificando en las Tablas 5.2 y 5.3 los parámetros que se usaron en el entrenamiento, que presentaron las mejores respuestas.

6.2.1 CNN VIS AlexNet/VGG16

La etapa de entrenamiento de estos dos clasificadores consistió en ajustar los pesos w_{L,n_ℓ} y coeficientes b_{L,n_ℓ} de algunas de las últimas capas de cada arquitectura, especificadas en la Sección 5.3.4, para utilizar la capacidad de *extraer características* de ambos modelos que fueron previamente entrenados con el banco de imágenes ImageNet [13].

Para realizar este *aprendizaje supervisado*, denominado algunas veces en la literatura como *ajuste fino* del modelo, se utilizaron los subconjuntos (5.6a) y (5.6b) del dataset Λ_v (5.6), cada uno de estos con $N_{\mathrm{T}_v} = 2680$ y $N_{\mathcal{V}_v} = N_{\mathrm{E}_v} = 335$ imágenes, distribuyendo las observaciones en un porcentaje del 50 % entre ambas clases.

Las Figuras 6.5 y 6.6 son las *curvas de aprendizaje*, que muestran el desarrollo de adiestramiento de cada red calculando la *exactitud* (4.9) y *pérdida* (3.33) en cada *epoch*. Los últimos valores obtenidos de estos registros son los siguientes

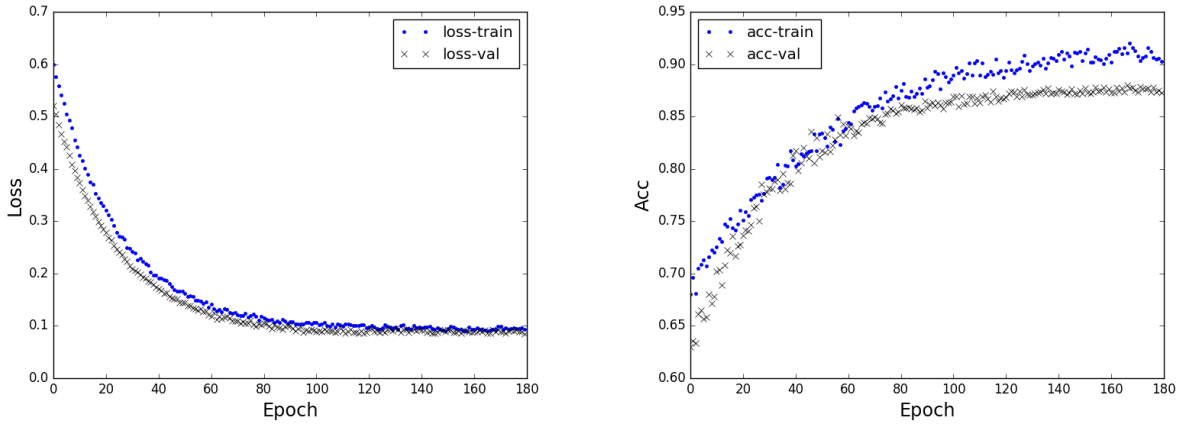
	Índice	
Dataset	$\mathcal{A}cc$	J
T_v	0.915	0.0953
$\mathcal{V}_v$	0.886	0.0821

(a) AlexNet.

	Índice	
Dataset	$\mathcal{A}cc$	J
T_v	0.93	0.0821
$\mathcal{V}_v$	0.904	0.0748

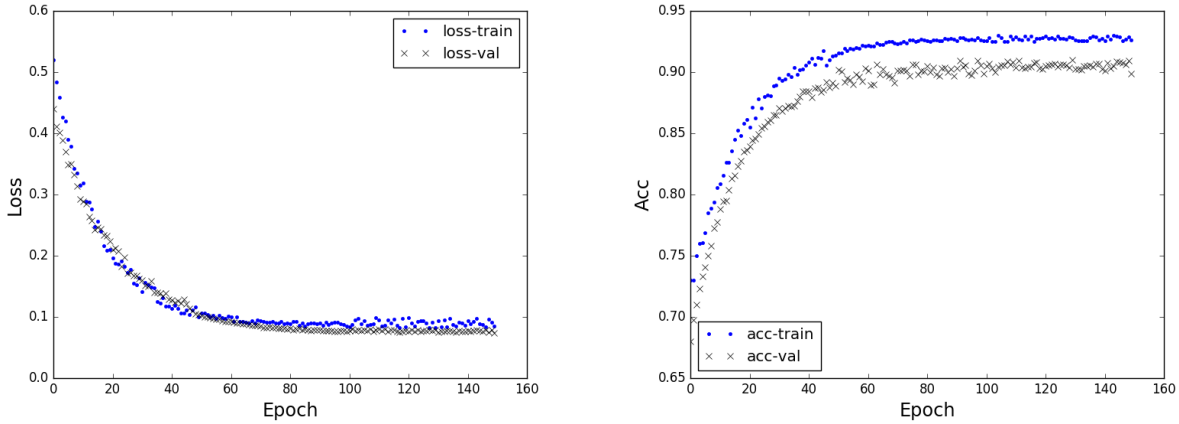
(b) VGG16.

Tabla 6.6. Índices finales de exactitud y precisión registrados del proceso de entrenamiento de los modelos propuestos CNN VIS.



(a) Registro de error. (b) Registro de exactitud.

Figura 6.5. Curvas del progreso de aprendizaje del modelo AlexNet.



(a) Registro de error. (b) Registro de exactitud.

Figura 6.6. Curvas del progreso de aprendizaje del modelo VGG16.

Utilizando la colección de imágenes E_v, se elaboraron las *matrices de contingencia* para cada clasificador, que se presentan a continuación

	Clase 1 $y' = 1$	Clase -1 $y' = -1$
Clase 1 $y = 1$	168	33
Clase 1 $y = -1$	14	120

(a) AlexNet.

	Clase 1 $y' = 1$	Clase -1 $y' = -1$
Clase 1 $y = 1$	172	10
Clase 1 $y = -1$	25	128

(b) VGG16.

Tabla 6.7. Matrices de confusión de los clasificadores CNN VIS.

Posteriormente se calcularon los indicadores de evaluación citados en la Sección 4.3 para cada una de las redes neuronales, con el objetivo de contrastar su capacidad de clasificación con los siguientes datos

Modelo	Exactitud	Precisión	Tasa de éxito	Medida $\mathbf{F_1}$
AlexNet	85.97 %	83.58 %	92.3 %	87.72 %
VGG16	89.55 %	94.5 %	87.3 %	90.76 %

Tabla 6.8. Indicadores de desempeño de los clasificadores

El índice $\mathbf{F_1}$ es un promedio ponderado entre la precisión y la tasa de éxito, que refleja el grado de balance entre los indicadores (4.10) y (4.11), determinado por la siguiente expresión

$$\mathbf{F_1} = 2 \cdot \frac{\mathcal{R} \cdot \mathcal{P}}{\mathcal{P} + \mathcal{R}} \tag{6.12}$$

6.2.2 CNN's IR-VIS

Para estos tres clasificadores se desarrolló su aprendizaje prescindiendo de un entrenamiento previo, utilizando los subconjuntos de la colección Λ_{iv} (5.7) que consta de $N_{\mathrm{T}_{iv}} = 2000$ y $N_{\mathcal{V}_{iv}} = N_{\mathrm{E}_{iv}} = 250$ imágenes, también considerando el 50 % de observaciones positivas y negativas.

Manteniendo la estructura de presentación de datos y resultados del apartado anterior, los últimos índices de *exactitud* y de la *función de costo*, del proceso de entrenamiento de cada red fueron los siguientes

	Índice	
Dataset	$\mathcal{A}cc$	J
T_v	0.902	0.129
$\mathcal{V}_v$	0.894	0.095

(a) HOG.

	Índice	
Dataset	$\mathcal{A}cc$	J
T_v	0.84	0.157
$\mathcal{V}_v$	0.803	0.135

(b) SIFT.

	Índice	
Dataset	$\mathcal{A}cc$	J
T_v	0.82	0.126
$\mathcal{V}_v$	0.794	0.107

(c) SURF.

Tabla 6.9. Índices finales de exactitud y precisión registrados del proceso de entrenamiento de los modelos propuestos CNN IR-VIS.

La memoria de cálculo de estos indicadores de evaluación se presenta en forma gráfica en las *curvas de aprendizaje*, de cada clasificador que se muestran en las Figuras 6.7, 6.8 y 6.9. Y utilizando la colección de imágenes E_{iv}, se elaboraron las *matrices de contingencia* para cada clasificador presentadas a continuación

	Clase 1 $y' = 1$	Clase -1 $y' = -1$
Clase 1 $y = 1$	119	9
Clase 1 $y = -1$	21	101

(a) HOG.

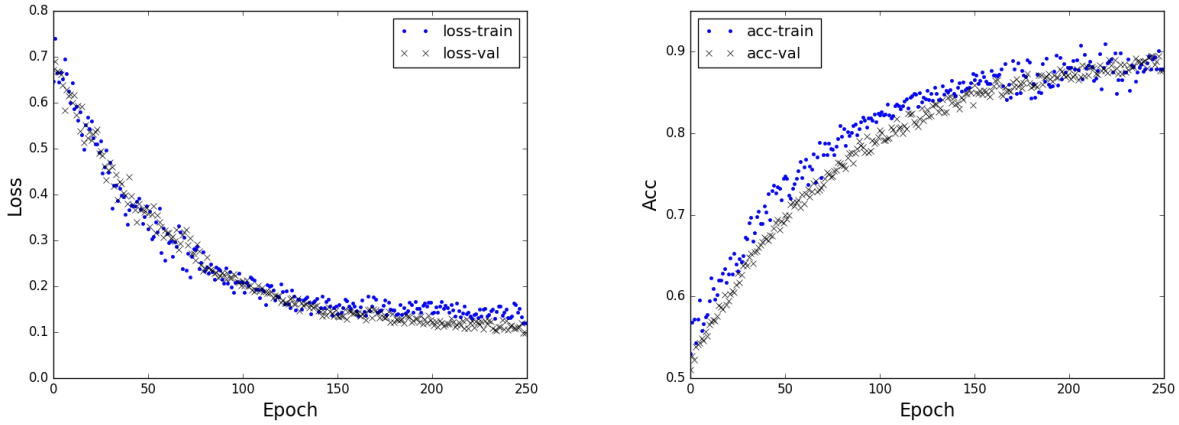
	Clase 1 $y' = 1$	Clase -1 $y' = -1$
Clase 1 $y = 1$	100	21
Clase 1 $y = -1$	24	105

(b) SIFT.

	Clase 1 $y' = 1$	Clase -1 $y' = -1$
Clase 1 $y = 1$	92	16
Clase 1 $y = -1$	38	104

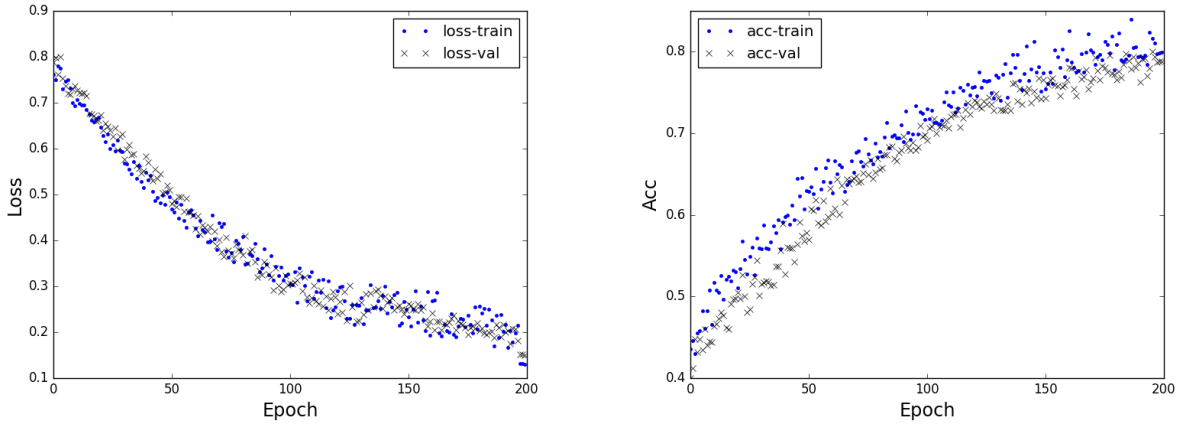
(c) SURF.

Tabla 6.10. Matrices de confusión de los clasificadores CNN IR - VIS.



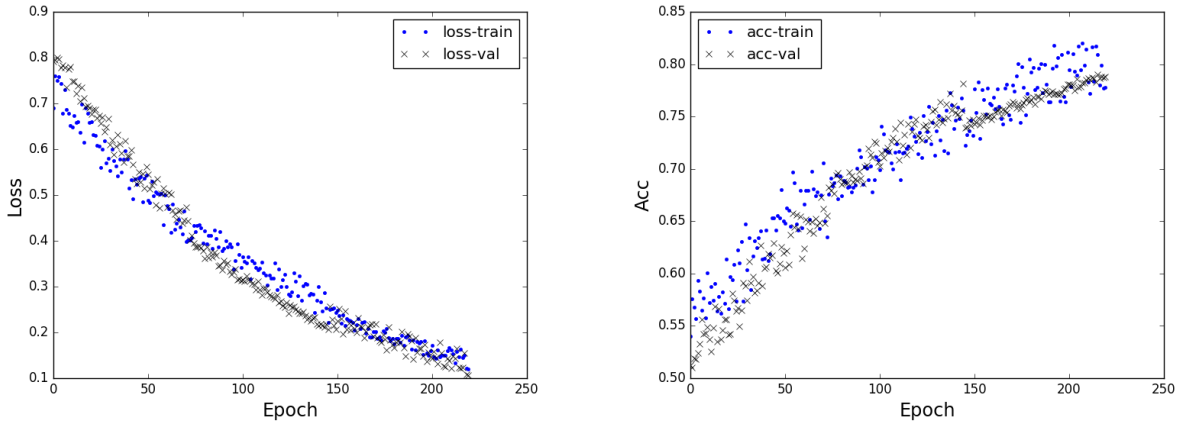
(a) Registro de error. (b) Registro de exactitud

Figura 6.7. Curvas del progreso de aprendizaje del modelo CNN IR-VIS HOG.



(a) Registro de error. (b) Registro de exactitud

Figura 6.8. Curvas del progreso de aprendizaje del modelo CNN IR-VIS SIFT.



(a) Registro de error.

(b) Registro de exactitud

Figura 6.9. Curvas del progreso de aprendizaje del modelo CNN IR-VIS SURF.

Y los índices obtenidos durante la evaluación fueron

Modelo	Exactitud	Precisión	Tasa de éxito	Medida $\mathbf{F_1}$
CNN IR-VIS HOG	88 %	92.96 %	85 %	88.8 %
CNN IR-VIS SIFT	82 %	82.64 %	80.64 %	81.63 %
CNN IR-VIS SURF	78.4 %	85.18 %	70.76 %	77.31 %

Tabla 6.11. Indicadores de desempeño de los clasificadores

Los índices calculados para comparar la respuesta de los clasificadores binarios desarrollados, se evaluaron considerando como ***Clase 1*** todas las observaciones que tuvieran una predicción mayor al 50 %, de lo contrario se catalogaba con la clase opuesta. que cada observación tuviera

6.3 Estimación de tiempos de procesamiento

Las dos cámaras transmiten vídeo a 30 imágenes por segundo, involucrando un período aproximado de 33.3 milisegundos para procesar las imágenes y proporcionar un resultado poder afirmar que el sistema funciona en tiempo real, sin embargo esta velocidad de transferencia puede disminuirse si la aplicación no demanda dicha rapidez.

Para reportar si el sistema de detección propuesto puede trabajar en tiempo real, se hicieron mediciones de la duración de los procesos principales y sus variaciones, utilizando el hardware de desarrollo mencionado en la Sección 5.3.5 y la biblioteca ***time*** de Python. De cada conjunto de lecturas se estimo un promedio y se registraron en la Tabla 6.12.

Proceso	*Tiempo para realizarse*
Rectificación de imágenes	6 ms
Búsqueda de ROI's en imagen IR	2 ms
Correspondencia IR → VIS	0.87 ms

(a) Procesos para obtener regiones de análisis.

Proceso	*Tiempo para realizarse*
Sin descriptor	0.5 ms
Descriptor HOG	380 ms
Descriptor SIFT	110 ms
Descriptor SURF	60 ms

(b) Formación de un tensor para ingresar al clasificador.

Proceso	*Tiempo para realizarse*
CNN VIS AlexNet	10 ms
CNN VIS VGG16	13 ms
CNN IR-VIS HOG	11 ms
CNN IR-VIS SIFT	10 ms
CNN IR-VIS SURF	12 ms

(c) Tiempo de respuesta de los clasificadores.

Tabla 6.12. Registros de duraciones de tiempo promedio de los procesos clave del sistema de detección propuesto.

Estas mediciones se hicieron considerando que las imágenes de las dos cámaras, ya estaban listas para utilizarse en los procesos listados, de los cuales todos los de la Tabla 6.12a son necesarios porque integran la **etapa de búsqueda de regiones** de las imágenes por analizar (denominadas como RoI) y en total suman un tiempo aproximado de **8.87 milisegundos**.

Después se agrega el tiempo de **preparación del tensor** que ingresa a cada una de las redes neuronales. En la Tabla 6.12b se observa que al utilizar descriptores de características, el tiempo de procesamiento aumenta considerablemente en comparación con solo estructurar alguna de las dos regiones de imagen en un arreglo.

Y por último las duraciones promedio de **propagación de un tensor**, en cada una de las CNN que se trabajaron se mantuvo relativamente constante, a pesar de la variación de los parámetros de cada arquitectura. Los tiempos registrados en la Tabla 6.12c incluyen la presentación del porcentaje de predicción en las imágenes.

Capítulo 7

Conclusiones

Después de experimentar y evaluar los clasificadores y diferentes procesos principales, que integran el sistema de detección de víctimas propuesto descrito en el capítulo 5 y presentado en la Figura 5.1, en seguida se exponen anotaciones, comentarios y conclusiones con base en los resultados reportados en el capítulo seis.

Al desarrollar las dos metodologías de correspondencia de pixeles entre las imágenes de las cámaras LWIR y VIS, se obtuvieron los parámetros de relación para ambos casos, logrando *errores de reproyección menores de un pixel* como se reportó en las Tablas 6.4 y 6.5.

De forma particular el método de *interpolación geométrica* [56] generó el menor error (2.29), logrando relacionar todos los puntos en las dos imágenes. En cambio la *diferencia de reproyección* lograda con el procedimiento de *geometría epiolar* fue mayor y solo correlacionó el 75 % de los pixeles de toda la resolución, reflejado al rectificar ambas imágenes en una, sin embargo, esta unión permitió obtener una visualización compuesta estilo estereoscópico con un tamaño de imagen razonable para inspecciones por observación.

La utilización de imágenes de diferentes perspectivas del *patrón de calibración* y sus imprecisiones de fabricación, posiblemente fueron la razones que más trascendieron en la rectificación y error de reproyección logrado con la relación de correspondencia definida por (6.8) y (6.7).

En consecuencia si se mejora el patrón de calibración, se utiliza otra cámara VIS de la misma calidad, pero que pueda estar más cerca del centro de la cámara LWIR y se hace más robusto el soporte de las cámaras, podría ser posible rectificar todos los pixels.

Al utilizar los vestigios de calor como guía de segmentación para determinar las *regiones de análisis* para el clasificador, se agilizó considerablemente el proceso de detección y también se obtiene una independencia de cambios dinámicos de iluminación y falta de contraste del entorno.

La estrategia de aprendizaje regional de R-CNN [29], contribuyó al análisis de proyecciones de personas parcialmente ocluidas, porque al evaluar cada uno de los clasificadores neuronales con imágenes complicadas como las mostradas en la Figura 6.12, se obtuvieron índices $\mathbf{F_1}$ muy semejantes a su exactitud de catalogación, de cada modelo.

Al utilizar la *transferencia de aprendizaje* con los modelos AlexNet y VGG16 se constato la relevancia de los entrenamientos profundos, porque con los reajustes hechos a los parámetros de estas redes, usando solo imágenes VIS, se obtuvieron índices de rendimiento (presentados en la Tabla 6.11) funcionales a pesar de la diferencia de las colecciones de imágenes usadas como ejemplo.

Con las redes CNN IR-VIS se obtuvieron índices de evaluación cercanos a las CNN VIS, considerando el número de muestras utilizadas para su entrenamiento y que este inició desde cero. La propuesta de experimentar con redes neuronales que procesan tensores formados por descriptores de características, tuvo como objetivo comprobar si la información captada por los diferentes sensores se lograba complementar, en un mismo espacio definido por los descriptores.

De acuerdo a los resultados de la Tabla 6.11, se puede decir que a nivel gráfico la integración de información IR y VIS, favorece a tener mejores resultados en un sistema de detección de personas, porque la red CNN IR-VIS HOG fue la que logró una mejor calificación de estos tres modelos combinados aunque, los porcentajes de exactitud y del índice $\mathbf{F_1}$ de las otras dos redes CNN IR-VIS no fueron tan lejanos.

Sin embargo, los niveles de precisión no superaron los obtenidos con las arquitecturas que procesan imágenes VIS, por este motivo se puede afirmar que el análisis de estas dos fuentes por separado como se proyecto en el capítulo 5, no brinda más beneficios que al solo usar espacios de colores. Pero si se obtiene rectificación estereoscópica completa RGB-LWIR, la información de textura se complementa y puede ser mejor que al solo usar imágenes VIS.

Por ello si es factible entrenar modelos de clasificación, utilizando imágenes RGB-LWIR, a expensa de hacer una colección de pares de imágenes numerosa y representativa de víctimas reales o simuladas. A esto se suma que el tiempo de respuesta del clasificador y etapas previas, fue menor al no utilizar descriptores de características y procesar directamente la imagen, registrando que incluso pueden ser capaces de trabajar con transmisiones de vídeo de 20 cuadros por segundo.

Presentar un resultado cada segundo en la aplicación de detección de víctimas, es aceptable porque se asume que las personas se encuentran en estado de reposo, por lo que se podría utilizar hasta la red CNN IR-VIS HOG que necesita en promedio medio segundo para su procesamiento. Pero si se desea una mayor fluidez, es posible llegar a utilizar una arquitectura entrenada con la información directa de imágenes rectificadas RGB-LWIR.

Referencias

[1] C. García, *Detección electrónica de víctimas no superficiales* (Facultad de Ingeniería, UNAM, 2016).

[2] A. Nezirović, *Trapped-victim detection in post-disaster scenarios using ultra-wideband radar*, Tesis de Ph.D., T.U. Delft, Delft University of Technology (2010).

[3] A. Kleiner and R. Kümmerle, *Genetic mrf model optimization for real-time victim detection in search and rescue.* International Conference on Intelligent Robots and Systems. IROS 2007. IEEE/RSJ , 3025 (2007).

[4] P. Viola and M. Jones, *Rapid object detection using a boosted cascade of simple features,* in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2001, CVPR 2001* (IEEE, 2001) pp. 511–518.

[5] A. Ben-Hur, D. Horn, H. T. Siegelmann, and V. Vapnik, *Support vector clustering,* Journal of machine learning research **2**, 125 (2001).

[6] S. Mallat, *A wavelet tour of signal processing: the sparse way* (Academic press, 2008).

[7] M. Andriluka, P. Schnitzspan, J. Meyer, S. Kohlbrecher, K. Petersen, O. Von Stryk, S. Roth, and B. Schiele, *Vision based victim detection from unmanned aerial vehicles, 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, **1**, 1740 (2010).

[8] M. Andriluka, S. Roth, and B. Schiele, *Pictorial structures revisited: People detection and articulated pose estimation,* in *IEEE Conference on Computer Vision and Pattern Recognition 2009* (IEEE, 2009) pp. 1014–1021.

[9] P. F. Felzenszwalb, D. A. McAllester, D. Ramanan, *et al.*, *A discriminatively trained, multiscale, deformable part model.* in *Cvpr*, Vol. 2 (2008) p. 7.

[10] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, *Object detection with discriminatively trained part-based models,* IEEE transactions on pattern analysis and machine intelligence **32**, 1627 (2010).

[11] M. A. Fischler and R. A. Elschlager, *The representation and matching of pictorial structures,* IEEE Transactions on computers **100**, 67 (1973).

[12] P. F. Felzenszwalb and D. P. Huttenlocher, *Pictorial structures for object recognition,* International journal of computer vision **61**, 55 (2005).

[13] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, *Imagenet large scale visual recognition challenge,* International journal of computer vision **115**, 211 (2015).

[14] B. Soni and A. Sowmya, *Classifier ensemble with incremental learning for disaster victim detection,* in *2012 IEEE international conference on robotics and biomimetics, ROBIO* (IEEE, 2012) pp. 446–451.

[15] T. Hastie, S. Rosset, J. Zhu, and H. Zou, *Multi-class adaboost,* Statistics and its Interface **2**, 349 (2009).

[16] T. Cover and P. Hart, *Nearest neighbor pattern classification,* IEEE transactions on information theory **13**, 21 (1967).

[17] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)* (Springer-Verlag, Berlin, Heidelberg, 2006).

[18] N. Dalal and B. Triggs, *Histograms of oriented gradients for human detection, international Conference on computer vision & Pattern Recognition (CVPR'05)*, **1**, 886 (2005).

[19] D. G. Lowe, *Distinctive image features from scale-invariant keypoints,* International journal of computer vision **60**, 91 (2004).

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, *Imagenet classification with deep convolutional neural networks,* in *Advances in neural information processing systems* (2012) pp. 1097–1105.

[21] I. A. Sulistijono and A. Risnumawan, *From concrete to abstract: Multilayer neural networks for disaster victims detection,* in *2016 International Electronics Symposium (IES)* (IEEE, 2016) pp. 93–98.

[22] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, *Decaf: A deep convolutional activation feature for generic visual recognition,* in *International conference on machine learning* (2014) pp. 647–655.

[23] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, *The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results,* http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html.

[24] K. E. Van de Sande, J. R. Uijlings, T. Gevers, and A. W. Smeulders, *Segmentation as selective search for object recognition,* in *IEEE International Conference on Computer Vision (ICCV), 2011* (IEEE, 2011) pp. 1879–1886.

[25] M. B. Bejiga, A. Zeggada, A. Nouffidj, and F. Melgani, *A convolutional neural network approach for assisting avalanche search and rescue operations with uav imagery,* Remote Sensing **9**, 100 (2017).

[26] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, *Going deeper with convolutions,* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015) pp. 1–9.

[27] W. Ouyang and X. Wang, *Joint deep learning for pedestrian detection,* in *Proceedings of the IEEE International Conference on Computer Vision* (2013) pp. 2056–2063.

[28] U. Michelucci, *Applied Deep Learning: A Case-Based Approach to Understanding Deep Neural Networks* (Apress, 2018).

[29] R. Girshick, J. Donahue, T. Darrell, and J. Malik, *Rich feature hierarchies for accurate object*

detection and semantic segmentation, in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2014) pp. 580–587.

[30] G. Shu, A. Dehghan, O. Oreifej, E. Hand, and M. Shah, *Part-based multiple-person tracking with partial occlusion handling,* in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on* (IEEE, 2012) pp. 1815–1821.

[31] P. Dollar, C. Wojek, B. Schiele, and P. Perona, *Pedestrian detection: An evaluation of the state of the art,* IEEE transactions on pattern analysis and machine intelligence **34**, 743 (2012).

[32] J. Wong, C. Robinson, *et al.*, *Urban search and rescue technology needs: identification of needs,* Federal Emergency Management Agency (FEMA) and the National Institute of Justice (NIJ). Document **207771** (2004).

[33] F. Suard, A. Rakotomamonjy, A. Bensrhair, and A. Broggi, *Pedestrian detection using infrared images and histograms of oriented gradients,* in *Intelligent Vehicles Symposium, 2006 IEEE* (IEEE, 2006) pp. 206–212.

[34] Q.-C. Pham, L. Gond, J. Begard, N. Allezard, and P. Sayd, *Real-time posture analysis in a crowd using thermal imaging,* in *2007 IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2007) pp. 1–8.

[35] Y. Yan, J. Ren, H. Zhao, G. Sun, Z. Wang, J. Zheng, S. Marshall, and J. Soraghan, *Cognitive fusion of thermal and visible imagery for effective detection and tracking of pedestrians in videos,* Cognitive Computation **10**, 94 (2018).

[36] S. J. Krotosky and M. M. Trivedi, *Person surveillance using visual and infrared imagery,* IEEE transactions on circuits and systems for video technology **18**, 1096 (2008).

[37] A. Y. Jiménez Rodríguez, *Análsis de vídeo térmico para la detección de signos vitales*, Tesis de Maestría, Universidad Nacional Autónoma de México, Posgrado de Ingeniería Eléctrica (2017).

[38] C. García, Y. Jiménez R A, and L. Escobar, *Thermal detection system for not superficial victims heartbeat detection,* International Journal of Information and Electronics Engineering **6**, 280 (2016).

[39] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision* (Cambridge University Press, 2003).

[40] S. J. Prince, *Computer vision: models, learning, and inference* (Cambridge University Press, 2012).

[41] W. Burger, *Zhang's Camera Calibration Algorithm: In-Depth Tutorial and Implementation*, Tech. Rep. HGB16-05 (University of Applied Sciences Upper Austria, School of Informatics, Communications and Media, Dept. of Digital Media, Hagenberg, Austria, 2016).

[42] R. Szeliski, *Computer vision: algorithms and applications* (Springer Science & Business Media, 2010).

[43] C. C. Slama, C. Theurer, and S. W. Henriksen, *Manual of photogrammetry* (American Society of photogrammetry, 1980).

[44] J. G. Fryer, *Camera calibration in non-topographic photogrammetry,* Handbook of Non Topographic Photogrammetry, American Society of Photogrammetry and Remote Sensing **2**, 51 (1989).

[45] A. E. Conrady, *Decentred lens-systems,* Monthly notices of the royal astronomical society **79**, 384 (1919).

[46] D. C. Brown, *Decentering distortion of lenses,* Photogrammetric Engineering and Remote Sensing (1966).

[47] J. G. Fryer and D. C. Brown, *Lens distortion for close-range photogrammetry,* Photogrammetric engineering and remote sensing **52**, 51 (1986).

[48] T. Melen, *Geometrical Modelling and Calibration of Video Cameras for Underwater Navigation*, ITK-rapport (Institutt for Teknisk Kybernetikk, Universitetet i Trondheim, Norges Tekniske Høgskole, 1994).

[49] J. Heikkila and O. Silven, *A four-step camera calibration procedure with implicit image correction,* in *1997 IEEE Proceedings of Computer Society Conference on Computer Vision and Pattern Recognition.* (IEEE, 1997) pp. 1106–1112.

[50] Z. Zhang, *A flexible new technique for camera calibration,* IEEE Transactions on pattern analysis and machine intelligence **22**, 1330 (2000).

[51] L. Quan and Z. Lan, *Linear n-point camera pose determination,* IEEE Transactions on pattern analysis and machine intelligence **21**, 774 (1999).

[52] R. Tsai, *A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses,* IEEE Journal on Robotics and Automation **3**, 323 (1987).

[53] Y. Abdel-Aziz, H. Karara, and M. Hauck, *Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry,* Photogrammetric Engineering & Remote Sensing **81**, 103 (2015).

[54] K. Levenberg, *A method for the solution of certain non-linear problems in least squares,* Quarterly of applied mathematics **2**, 164 (1944).

[55] D. W. Marquardt, *An algorithm for least-squares estimation of nonlinear parameters,* Journal of the society for Industrial and Applied Mathematics **11**, 431 (1963).

[56] J. H. Lee, J.-S. Choi, E. S. Jeon, Y. G. Kim, T. T. Le, K. Y. Shin, H. C. Lee, and K. R. Park, *Robust pedestrian detection by combining visible and thermal infrared cameras,* Sensors **15**, 10580 (2015).

[57] F. B. Campo, F. L. Ruiz, and A. D. Sappa, *Multimodal stereo vision system: 3d data extraction and algorithm evaluation,* IEEE Journal of Selected Topics in Signal Processing **6**, 437 (2012).

[58] C. Wang, Y. K. Cho, and M. Gai, *As-is 3d thermal modeling for existing building envelopes using a hybrid lidar system,* Journal of Computing in Civil Engineering **27**, 645 (2012).

[59] Y. Chen and W. Warren, *3d fusion of infrared images with dense rgb reconstruction from multiple views-with application to fire-fighting robots,* Recuperado a partir de https://pdfs. semanticscholar. org/9702/31551cbf43023bb77d0a7c233b5311c31997. pdf (2013).

[60] J.-Y. Bouguet, *Camera calibration toolbox for matlab,* (2015).

[61] J.-Y. Bouguet *et al.*, *Visual methods for three-dimensional modeling* (Citeseer, 1999).

[62] H. C. Longuet-Higgins, *A computer algorithm for reconstructing a scene from two projections,* Nature **293**, 133 (1981).

[63] E. Fendri, R. R. Boukhriss, and M. Hammami, *Fusion of thermal infrared and visible spectra for robust moving object detection,* Pattern Analysis and Applications **20**, 907 (2017).

[64] J. Ma, Y. Ma, and C. Li, *Infrared and visible image fusion methods and applications: a survey,* Information Fusion **45**, 153 (2019).

[65] R. Gonzalez and R. Woods, *Digital Image Processing* (Pearson/Prentice Hall, 2008).

[66] M. S. Nixon and A. S. Aguado, *Feature extraction & image processing for computer vision* (Academic Press, 2012).

[67] J. M. Prewitt, *Object enhancement and extraction,* Picture processing and Psychopictorics **10**, 15 (1970).

[68] J. Prewitt and M. L. Mendelsohn, *The analysis of cell images,* Annals of the New York Academy of Sciences **128**, 1035 (1966).

[69] I. Sobel, *History and definition of the so-called sobel operator,* (2014).

[70] H. Scharr, *Optimale Operatoren in der digitalen Bildverarbeitung*, Tesis de Ph.D. (2000).

[71] J. Canny, *A computational approach to edge detection,* IEEE Transactions on pattern analysis and machine intelligence , 679 (1986).

[72] D. Marr and E. Hildreth, *Theory of edge detection,* Proc. R. Soc. Lond. B **207**, 187 (1980).

[73] I. R. Otero, *Anatomy of the SIFT Method*, Tesis de Ph.D., École normale supérieure de Cachan-ENS Cachan (2015).

[74] H. Bay, T. Tuytelaars, and L. Van Gool, *Surf: Speeded up robust features,* in *European conference on computer vision* (Springer, 2006) pp. 404–417.

[75] E. Oyallon and J. Rabin, *An analysis of the surf method,* Image Processing On Line **5**, 176 (2015).

[76] C. C. Aggarwal, *Neural networks and deep learning* (Springer, 2018).

[77] N. Buduma and N. Locascio, *Fundamentals of deep learning: Designing next-generation machine intelligence algorithms* (O' Reilly Media, Inc., 2017).

[78] S. Haykin, *Neural networks and learning machines*, Vol. 3 (Pearson Upper Saddle River, 2009).

[79] K. P. Murphy, *Machine Learning: A Probabilistic Perspective. Adaptive Computation and Machine Learning* (MIT press, 2012).

[80] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, 2016) `http://www.deeplearningbook.org`.

[81] F. Chollet, *Deep learning with python* (Manning Publications Co., 2017).

[82] K. Simonyan and A. Zisserman, *Very deep convolutional networks for large-scale image recognition,* arXiv preprint arXiv:1409.1556 (2014).

[83] P. Flach, *Machine learning: the art and science of algorithms that make sense of data* (Cambridge University Press, 2012).

[84] A. Kaehler and G. Bradski, *Learning OpenCV 3: computer vision in C++ with the OpenCV library* (O' Reilly Media, Inc., 2016).

[85] S. Shalev-Shwartz and S. Ben-David, *Understanding machine learning: From theory to algorithms* (Cambridge university press, 2014).

[86] R. Girshick, *Fast r-cnn,* in *Proceedings of the IEEE international conference on computer vision* (2015) pp. 1440–1448.

[87] S. Ren, K. He, R. Girshick, and J. Sun, *Faster r-cnn: Towards real-time object detection with region proposal networks,* in *Advances in neural information processing systems* (2015) pp. 91–99.

[88] R. Girshick, F. Iandola, T. Darrell, and J. Malik, *Deformable part models are convolutional neural networks,* in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (2015) pp. 437–446.

[89] OPTRIS, *Basic Principles of non-contact temperature measurement* (Optris GmbH, 2010).

[90] J. Wagner, V. Fischer, M. Herman, and S. Behnke, *Multispectral pedestrian detection using deep fusion convolutional neural networks,* in *24th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)* (2016) pp. 509–514.

[91] C. Poynton, *Digital video and HD: Algorithms and Interfaces* (Elsevier, 2012).

Lista de Acrónimos y siglas

2D Two Dimensions. 12, 14, 17, 18, 20, 22, 30, 31, 34, 39, 60, 68, 76, 86, 87

3D Three Dimensions. 12, 14, 17–22, 26, 27, 31, 83

AP Average Precision. 78

AUC Area Under Curve. 78

CNN Convolutional Neural Network. 4, 5, 51, 55, 60–63, 68, 73–75, 77, 88, 89, 93, 101, 103, 105, 108

DLT Direct Linear Transformation. 21, 22

DoG Diferencia de Gaussianas. 38

DPM Deformable Parts Model. 3, 5, 67, 71–75, 79

HOG Histogram of Oriented Gradients. 3, 39, 40, 71, 86, 87

ILSVRC Imagenet Large Scale Visual Recognition Competition. 3–6, 61–63, 73

IR Infrared Radiation. 6–9, 11, 23, 26, 30, 31, 65, 81, 83, 84, 86–90, 93, 94, 96, 98–100, 103, 108

k-NN k-Nearest Neighbors. 3, 67

LM Levenberg-Marquart algorithm. 22, 25

LoG Laplaciano del Gaussiano. 38, 48

LWIR Long Wave Infrared Radiation. 7–9, 30, 31, 81–84, 94, 107, 108

mAP mean Average Precision. 78

NN Neural Network. 4, 51, 53–55, 59

PS Picture Structures. 3, 5, 69, 70, 72, 79

R-CNN Regional Convolutional Neural Network. 6, 73–75, 79, 85, 89, 91, 108

ROC Receiver Operating Characteristic. 78

RoI Region of Interest. 81, 82, 87, 105

ROS Robot Operating System. 82, 91, 94, 96

SIFT Scale-Invariant Feature Transform. 42, 44, 45, 48, 49, 72, 86, 87

SURF Speed-Up Robust Features. 3, 45, 48, 51, 86, 87

SVD Singular Value Decomposition. 21, 29, 98

SVM Support Vector Machine. 3, 5, 40, 68, 71, 73, 74

UAV Unmanned Aerial Vehicle. 3, 5

USAR Urban Search and Rescue. 1, 2, 7

VIS Visible Interval Spectrum. 7–9, 11, 23, 26, 30, 31, 65, 82–90, 93, 94, 96, 98–101, 103, 107, 108

VOC Visual Object Classes Challenge. 4, 79, 85, 88

Glosario

aprendizaje automático Área de la inteligencia artificial, dedicada a la creación de programas de computadora para solucionar diversas tareas, por medio de la formación de modelos de información determinados por la exposición o presentación de muestras ejemplares vinculadas con las respuestas que se desean obtener [79].. 2, 10, 51, 59, 66, 68, 75, 79, 88, 91

bounding box Perímetro de un cuadrilítero, que delimita cierta región de una imagen y la señala en esta misma.. 3, 30, 31, 68, 71, 73–75, 81, 82, 85, 89

campo de visión También denominado campo de perspectiva, es la extensión del mundo observable que es visto en algún momento. En referencia a instrumentos ópticos y sensores de las cámaras, son los ángulos horizontal y vertical a través de los cuales detecta la radiación electromagnética.. 7, 8, 11, 26–28, 30, 65, 82, 91, 94

dataset Colección estructurada de datos integrada por uno o más campos de información, que pueden correlacionarse entre sí por tendencias estadísticas o que son anotaciones de diferentes muestras de un conjunto representativo para obtener algún patrón.. 4, 5, 61, 73, 79, 85–91

entropía término de la *Teoría de la Información*, que hace referencia a una medida de incertidumbre de una variable aleatoria, con una distribución definida. Es utilizada para cuantificar la diferencia entre dos distribuciones de probabilidad y tiene por unidades los ***nats*** o ***bits***, en función de la base empleada en el logaritmo para su cálculo [80]. . 55, 56

epoch Etapa de presentación de las muestras de entrenamiento, durante el proceso de aprendizaje de la red. Después de este período, convencionalmente se tienden a reajustar la combinación de *pesos sinápticos* y *coeficientes de ajuste* [76].. 57, 58, 77, 90, 100

grafo Diagrama que representa mediante puntos y líneas las relaciones entre pares de elementos y que se usa para resolver problemas lógicos, topológicos y de cálculo combinatorio. Las redes neuronales comúnmente son expresadas en este tipo de diagramas. 3, 53, 54, 67, 69, 70, 75

hiper parámetro Es una variable involucrada en el proceso de *aprendizaje* de una red neuronal, aportando al nivel de funcionalidad del modelo entrenado. Algunos de los hiper parámetros son; tasa de aprendizaje γ, muestras de entrenamiento k, coeficientes de regulación β, número de *epochs* η, entre otros [80].. 58, 59, 90

middleware Software para el intercambio de información entre programas que asisten alguna aplicación, que brinda la capacidad de interactuar y comunicar programas en diferentes lenguajes. 82, 91

pixel Elemento mínimo entero de una imagen.. 4, 5, 8, 34, 36, 37, 40, 65, 66, 73–75, 81, 85, 91, 93, 94

RGB Espacio de color formado por los tres componentes que pueden representar el intervalo del espectro electromagnético visible, ante el ojo humano [91].. 5, 74, 108

sinapsis Conexión entre el axón de una neurona y la dendrita de otra cercana mediante neurotransmisores, que representa la relación funcional de dos neuronas.. 52

tensor Nombre utilizado en aplicaciones de aprendizaje automático para hacer referencia a los arreglos o estructuras de datos de n-dimensiones [81].. 52, 54, 74, 87, 105

transfer learning Concepto que hace referencia al uso de los pesos de una arquitectura de red neuronal, que fueron obtenidos normalmente, por un proceso de entrenamiento utilizando alguno de los *dataset* cánones, como Imagenet o PASCAL VOC.. 4, 63, 86, 89

yaml Es un formato estructurado de datos legible por humanos, utilizado comúnmente para archivos de configuración, o almacenamiento de datos resultantes de un proceso o que se transmiten, por mencionar algunos.. 96

YCbCr Familia de espacio de color usada en sistemas de vídeo y fotografía digital. **Y** representa la componente *luma*, los canales **Cb** y **Cr** son los componentes de diferencia de crominancia de azul y rojo respectivamente [91].. 2, 5, 74