Visión Artificial: Fundamentos, Técnicas y Aplicaciones

Daniel Santiago Ahumada - dsantiag7@cuc.edu.co

Isaac Montes - imontes2@cuc.edu.co

Ingeniería Electrónica

Propósito y alcance de la visión artificial

La visión artificial es un campo de la inteligencia artificial que busca que las máquinas puedan "ver" e interpretar imágenes digitales de manera análoga a la visión humana[1]. Esto implica desarrollar métodos para adquirir, procesar y analizar imágenes del mundo real y extraer de ellas información útil. Su objetivo principal es lograr que un sistema informático reconozca automáticamente patrones o características complejas presentes en las imágenes de diversos dominios[2]. En otras palabras, la visión artificial pretende que los computadores perciban y comprendan el contenido de una imagen o secuencia de imágenes y tomen decisiones basadas en esa comprensión, emulando la capacidad visual humana.

Procesamiento de imágenes vs. visión por computadora: roles y diferencias

La visión por computadora (o visión artificial) y el procesamiento de imágenes son conceptos relacionados pero distintos. El procesamiento de imágenes se enfoca en técnicas para tratar y mejorar imágenes: por ejemplo, filtrado, ajuste de contraste o transformaciones que modifican la imagen para mejorar su calidad visual o extraer información básica de ella[3]. Por su parte, la visión por computadora va más allá de la mejora de la imagen en sí; su meta es que las máquinas comprendan el contenido de las imágenes y puedan actuar en función de ese contenido[4]. Esto significa que la visión por computadora abarca tareas de alto nivel como reconocimiento de objetos, detección de eventos, estimación de escenas 3D, etc., combinando técnicas de procesamiento de imágenes con algoritmos de inteligencia artificial para interpretar lo que aparece en la imagen. En resumen, el procesamiento de imágenes se ocupa de manipular imágenes (suele ser un paso previo), mientras que la visión por computadora busca interpretar las imágenes y extraer conocimiento de ellas de forma automática.

Análisis de histogramas: interpretación y uso práctico

El histograma de una imagen es una representación gráfica que muestra la distribución de frecuencias de los niveles de intensidad (tonos de gris) o colores de los píxeles en dicha imagen[5]. Típicamente, el histograma se representa como un gráfico de barras donde el eje horizontal corresponde a los posibles valores de intensidad (de negro a blanco, o de 0

a 255 en una imagen de 8 bits) y el eje vertical indica la cantidad de píxeles que tienen cada valor. Este gráfico ofrece información valiosa sobre el brillo y el contraste de la imagen, mostrando si está predominando los tonos oscuros, claros o medios. Por ejemplo, un histograma muy concentrado en la zona izquierda indica una imagen oscura, mientras que uno repartido uniformemente sugiere buen contraste. Los histogramas se utilizan en visión artificial y fotografía para analizar y ajustar imágenes: permiten detectar problemas de exposición y aplicar correcciones (como ecualización de histograma) para mejorar la calidad visual[6].

Técnicas de filtrado para supresión de ruido (mediana y gaussiano)

Para reducir el ruido en una imagen (es decir, las variaciones indeseadas de color o brillo en los píxeles), se emplean filtros de suavizado. Uno de los más conocidos es el filtro de mediana, que sustituye cada píxel por el valor *mediano* de sus píxeles vecinos dentro de una ventana definida[7]. Este filtro es muy eficaz eliminando el ruido impulsivo tipo "sal y pimienta" (píxeles aleatoriamente blancos o negros) porque la mediana descarta los valores atípicos; además tiende a preservar mejor los bordes que un promedio simple. Otro filtro común es el filtro gaussiano, el cual realiza un promedio ponderado de cada píxel con sus vecinos según una distribución Gaussiana (en forma de campana)[8]. En la práctica, se aplica una convolución con una máscara Gaussiana, dando más peso al píxel central y reduciendo progresivamente la influencia de los píxeles más lejanos. El resultado es una imagen suavizada donde el ruido de alta frecuencia queda atenuado (menos "grano"), aunque a costa de un ligero desenfoque de los detalles finos.

Detección de contornos: operadores por gradiente y métodos avanzados

Entre las técnicas más utilizadas para la detección de bordes en imágenes se encuentran los operadores basados en el cálculo de diferencias o gradientes de intensidad. Un grupo importante son los operadores de primer orden, como los filtros de *Sobel*, *Prewitt* o *Roberts*, que aplican máscaras para aproximar la primera derivada (gradiente) en direcciones horizontal y vertical, resaltando las zonas con cambios bruscos de brillo. También se emplean operadores de segundo orden como el *Laplaciano*, que detecta bordes buscando cambios abruptos en la segunda derivada de la intensidad, y variantes suavizadas de este como el *Laplaciano del Gaussiano* (LoG, también conocido por el operador de Marr-Hildreth) donde se aplica un filtrado Gaussiano antes del cálculo laplaciano para reducir ruido. Una técnica muy difundida es el algoritmo de *Canny*, que combina varias etapas (suavizado Gaussiano, cálculo de gradiente, supresión de máximos no locales y umbralización con histéresis) para obtener bordes finos y bien conectados. En resumen, métodos populares incluyen los operadores de gradiente simples (Sobel, Prewitt), el operador Laplaciano/LoG y el detector de Canny, entre otros[9], cada uno con sus ventajas en cuanto a simplicidad, precisión y robustez al ruido.

Segmentación de imágenes: enfoques y su importancia en el flujo de trabajo

Un algoritmo de segmentación de imágenes es un procedimiento que divide una imagen digital en regiones o segmentos más pequeños, agrupando píxeles que comparten ciertas características homogéneas (como color, intensidad o textura)[10]. El resultado de la segmentación es típicamente un conjunto de áreas delineadas (o una "etiqueta" asignada a cada píxel indicando a qué región pertenece) tales que cada región corresponde a una parte significativa de la imagen, por ejemplo un objeto o un fondo. Esta tarea es crucial porque simplifica y estructura la representación de la imagen, facilitando su análisis: es importante ya que permite localizar regiones con significado dentro de la escena[11]. Al separar los objetos del fondo o distinguir diferentes componentes en la imagen, la segmentación prepara el terreno para pasos posteriores de visión artificial, como el reconocimiento de objetos (operando sobre cada región segmentada), el cómputo de características específicas por región (áreas, formas, texturas) o el seguimiento de objetos a lo largo de una secuencia de imágenes. En aplicaciones prácticas, una buena segmentación ayuda, por ejemplo, a que un sistema médico identifique un tumor aislándolo del tejido sano circundante, o a que un vehículo autónomo distinga la calzada, las señales y los peatones en su campo de visión.

SIFT: detección y descripción invariante a escala

El método SIFT (*Scale-Invariant Feature Transform*) es un algoritmo de visión por computadora diseñado para detectar y describir puntos de interés notables en una imagen, de forma que la descripción resulte invariante a cambios de escala y a rotaciones[12]. En primer lugar, SIFT localiza puntos clave (*keypoints*) distintivos encontrando máximos y mínimos locales en una representación en múltiples escalas de la imagen (usando diferencias de Gauss para generar un espacio de escala)[13]. Estos puntos tienden a corresponder con esquinas, extremos de bordes u otras estructuras locales bien definidas. A cada *keypoint* se le asigna luego una orientación dominante basada en la distribución de gradientes en su vecindad, lo que normaliza su dirección. Finalmente, el método construye un descriptor para cada punto de interés: básicamente un vector numérico (de 128 dimensiones en la versión estándar) que resume los patrones de orientación de los gradientes locales alrededor del punto[14]. Este descriptor –frecuentemente descrito como un histograma de orientaciones de gradiente– caracteriza de manera única la apariencia local en torno al punto clave. Gracias a este proceso, las características SIFT de una imagen pueden compararse con las de otra para encontrar coincidencias. SIFT es valioso en reconocimiento de objetos porque permite emparejar características locales entre imágenes diferentes de un mismo objeto aun cuando haya cambios de escala, rotación o iluminación, facilitando tareas como el reconocimiento o la reconstrucción de escenas[15].

SURF: características robustas y aceleradas para tiempo real

El método SURF (*Speeded-Up Robust Features*) es un algoritmo de detección y descripción de características locales que fue propuesto como una versión optimizada (más *rápida*) del SIFT. Al igual que SIFT, SURF identifica puntos de interés en la imagen y genera descriptores invariantes a escala y rotación, pero lo hace mediante técnicas que reducen el tiempo de cómputo. En particular, SURF emplea imágenes integrales y *filtros de caja* (aproximaciones rectangulares) en lugar de convoluciones gaussianas para calcular rápidamente las diferencias de intensidad, construyendo un espacio de escala de forma eficiente[16]. El detector de puntos de interés de SURF está basado en la matriz Hessiana: busca máximos locales del determinante de la Hessiana en distintas escalas, lo cual permite encontrar *blobs* (estructuras tipo mancha) que señalan características distintivas en la imagen[17]. Esta estrategia, conocida como Fast Hessian, acelera la detección manteniendo robustez y precisión en la localización de puntos. Una vez detectados los *keypoints*, SURF genera para cada uno un descriptor compacto (generalmente de 64 dimensiones, más corto que el de SIFT) basado en la distribución de orientaciones de los gradientes en su entorno. En conjunto, SURF logra una detección y descripción de características locales mucho más veloz que SIFT[18], conservando la robustez ante cambios de escala, rotación e iluminación de manera similar al algoritmo original[19]. Por ello es adecuado para aplicaciones en tiempo real o dispositivos con recursos limitados, manteniendo buen desempeño en tareas de emparejamiento y reconocimiento de objetos.

Transformada de Hough: detección de formas geométricas

La transformada de Hough es una técnica utilizada en visión artificial para detectar figuras geométricas especificadas por parámetros matemáticos (por ejemplo líneas rectas, circunferencias, elipses) dentro de una imagen digital[20]. Su principio fundamental es transformar los puntos de la imagen (frecuentemente puntos detectados como bordes) a un espacio de parámetros donde cada tipo de figura se representa de forma característica. Por ejemplo, en el caso de detección de líneas rectas, cada punto en la imagen puede transformarse a una familia de líneas (representadas en coordenadas de pendiente-intersección o en coordenadas polares ρ-θ) que pasarían por ese punto. Mediante un esquema de votación, la transformada de Hough acumula evidencias: cada punto de borde vota por todos los parámetros de figuras (líneas, círculos, etc.) que podrían atravesarlo, incrementando un contador en un acumulador de parámetros. Tras procesar todos los puntos, aquellos valores de parámetros que recibieron muchos votos señalan la presencia de una figura geométrica con esos parámetros en la imagen[21]. Gracias a este mecanismo, la transformada de Hough puede detectar de forma robusta líneas o curvas incluso si los bordes están fragmentados o hay ruido, pues logra agrupar múltiples píxeles dispersos que conjuntamente definen una misma forma. Es especialmente popular para detectar líneas en imágenes de carreteras, bordes de objetos, o círculos (p. ej. detección

de pupilas, monedas, etc.), proporcionando una forma de reconocer componentes geométricos básicos en la escena.

Relevancia de los bordes en la representación y análisis de escenas

La detección de bordes consiste en encontrar en una imagen los puntos donde ocurren cambios abruptos de intensidad, es decir, las fronteras entre distintas regiones o objetos. En términos prácticos, un detector de bordes identifica los píxeles que corresponden a contornos o límites de elementos dentro de la imagen[9]. Esto se logra calculando derivadas o diferencias locales en la intensidad: allí donde la intensidad varía bruscamente, se marca un borde. La detección de bordes es fundamental en visión artificial porque los bordes delinean la forma de los objetos y estructuras presentes, proporcionando una representación simplificada pero informativa de la escena. Al extraer los contornos principales, se reduce la cantidad de datos a procesar pero se conserva la información geométrica esencial, lo que facilita etapas posteriores como la segmentación de la imagen en regiones, el reconocimiento de objetos o la estimación de la forma y posición de objetos en el espacio[22]. En resumen, los bordes actúan como características de alto nivel que permiten separar objetos del fondo, identificar regiones de interés y servir de entrada para algoritmos más complejos; por ello, detectar bordes con precisión mejora significativamente la capacidad de un sistema de visión artificial para analizar y comprender imágenes.

Comparativa práctica: filtros Sobel frente al detector Canny

El filtro de **Sobel** es un operador clásico para detección de bordes que calcula una aproximación al gradiente (derivada primera) de la imagen. Se aplican dos convoluciones con máscaras de 3×3 –una en dirección horizontal y otra en vertical– conocidas como máscaras Sobel, que resaltan las variaciones de intensidad en cada eje. Esto produce dos imágenes (gradiente en X y en Y); combinándolas se obtiene la magnitud del gradiente en cada píxel, que indica la presencia de un posible borde[23]. El filtro de Sobel tiende a resaltar bien los bordes orientados vertical u horizontalmente y es sencillo de implementar; sin embargo, suele producir bordes algo gruesos y responde a ruido si la imagen no ha sido suavizada previamente.

El detector de **Canny** es un método más sofisticado y robusto para detección de bordes, compuesto por múltiples etapas secuenciales[24]. Primero realiza un suavizado Gaussiano de la imagen para reducir el ruido. Luego calcula el gradiente de intensidad (frecuentemente usando Sobel u operador similar) para obtener una imagen de magnitudes de borde. Sobre esa imagen aplica una supresión de no-máximos, que consiste en eliminar cualquier píxel que no sea el valor máximo de gradiente en su vecindad a lo largo de la dirección del gradiente; así se afinan los contornos dejando los bordes con un grosor de un solo píxel[25]. Finalmente, Canny utiliza una umbralización con histéresis: define dos umbrales, uno alto y uno bajo. Los píxeles de borde con valor de gradiente por

encima del umbral alto se marcan inmediatamente como bordes “fuertes”; los píxeles por debajo del umbral bajo se descartan; y aquellos con valores intermedios (entre ambos umbrales) se marcan como bordes solo si están conectados a algún borde fuerte. Este proceso de histéresis permite conservar bordes débiles reales conectados a bordes fuertes y descartar detecciones aisladas que probablemente sean ruido. El resultado del algoritmo Canny es una imagen binaria con bordes bien definidos, continuos y delgados, logrando una detección de contornos más limpia y precisa que la obtenida con operadores simples como Sobel.

Aprendizaje supervisado vs. no supervisado en aplicaciones visuales

En aprendizaje supervisado, el sistema de visión artificial se entrena utilizando datos de entrada junto con las respuestas deseadas (etiquetas) para esos datos. Es decir, cada imagen del conjunto de entrenamiento viene anotada con la categoría, objeto o valor que se espera que el sistema produzca como salida[26]. Durante el entrenamiento, el modelo compara sus predicciones con las etiquetas reales y ajusta sus parámetros para minimizar el error; de este modo “aprende” a asociar las características de la imagen con la etiqueta correcta. Un ejemplo sería entrenar un clasificador de imágenes proporcionándole muchas fotos de perros y gatos, donde cada foto está etiquetada como “perro” o “gato”: el algoritmo ajusta sus pesos internos hasta lograr distinguirlas con alta precisión. En cambio, en el aprendizaje no supervisado el sistema recibe únicamente las imágenes de entrada sin ninguna etiqueta o respuesta esperada. El objetivo en este caso es que el algoritmo descubra por sí mismo alguna estructura subyacente en los datos[27]. Por ejemplo, un método no supervisado puede agrupar imágenes según similitud (clústeres) o encontrar componentes comunes en las imágenes, sin saber de antemano qué representa cada grupo; un caso práctico en visión artificial es la segmentación automática en la que el algoritmo separa una imagen en regiones coherentes sin que alguien le haya indicado cuáles son “objeto” y “fondo”. En resumen, la diferencia principal es la presencia o ausencia de etiquetas durante el entrenamiento[28]. El aprendizaje supervisado aprovecha conocimientos previos (etiquetas) para alcanzar soluciones muy precisas en tareas definidas, mientras que el no supervisado busca patrones ocultos en los datos sin guía externa, lo cual es útil para exploración de datos o cuando la obtención de etiquetas es costosa o impracticable.

Características locales: descriptores y su papel en el reconocimiento de objetos

Las características locales de una imagen son rasgos o descriptores que corresponden a patrones distintivos en regiones pequeñas de la imagen, típicamente alrededor de puntos de interés bien definidos (como esquinas, esquinas de contorno o texturas sobresalientes). A diferencia de las características globales (que describen la imagen en su conjunto), las características locales se enfocan en detalles específicos de áreas limitadas. Un ejemplo son los *keypoints* detectados por algoritmos como SIFT o SURF: cada *keypoint* marca

una posición en la imagen con contenido único en su vecindad, y se le asocia un vector descriptor que captura la apariencia alrededor de ese punto. Estas descripciones suelen diseñarse para ser invariantes a transformaciones como cambios de escala, rotación o variaciones de iluminación, de modo que el mismo punto físico del objeto pueda reconocerse en distintas imágenes[29]. Las características locales son importantes para el reconocimiento de objetos porque actúan como "huellas digitales" de porciones del objeto. Permiten comparar y hacer *match* entre partes de dos imágenes: si varias características locales de un objeto en una imagen coinciden con las de otro objeto en otra imagen, es muy probable que se trate del mismo objeto. Esto hace posible identificar un objeto aunque esté en diferente posición o bajo distintas condiciones, ya que la coincidencia de múltiples características locales robustas confirma la presencia del objeto. En la época previa al aprendizaje profundo, muchos sistemas de reconocimiento se basaban en la detección de características locales (esquinas, puntos Harris, descriptores SIFT/SURF, etc.) para luego emparejarlas entre la imagen de consulta y modelos almacenados. Incluso en la era actual de las CNN, el concepto persiste: las redes convolucionales extraen automáticamente características locales en sus primeras capas[30], construyendo a partir de ellas representaciones más complejas. En resumen, las características locales proporcionan información discriminativa y *reutilizable* sobre fragmentos de un objeto, lo que es esencial para reconocer objetos de forma independiente a la escala, la rotación o el fondo donde aparezcan.[29]

Bibliografía

[1] [2] Visión artificial - Wikipedia, la enciclopedia libre

https://es.wikipedia.org/wiki/Visi%C3%B3n_artificial

[3] [4] Visión artificial, procesamiento de imágenes y computer vision

https://secmotic.com/vision-artificial-procesamiento-de-imagenes-y-computer-vision/

[5] [6] Curso de Procesado de Imagen (c)GPI

https://www.uv.es/gpoei/eng/Pfc_web/realzado/histograma/histo.htm

[7] [8] Microsoft PowerPoint - tema3-1.ppt [Modo de compatibilidad]

https://grupo.us.es/gtocoma/pid/tema3-1.pdf

[9] PyPro | Ciencia de Datos

https://www.pypro.mx/app/curso/vision-artificial-con-opencv/deteccion-de-bordes-y-segmentacion-de-imagenes

[10] Procesamiento digital de imágenes y su importancia - MasScience

https://www.masscience.com/procesamiento-digital-de-imagenes-y-su-importancia/

[11] Segmentación (procesamiento de imágenes) - Wikipedia, la enciclopedia libre

https://es.wikipedia.org/wiki/Segmentaci%C3%B3n_(procesamiento_de_im%C3%A1genes)

[12] [13] [15] [30] Scale-invariant feature transform - Wikipedia, la enciclopedia libre

https://es.wikipedia.org/wiki/Scale-invariant_feature_transform

[14] [PDF] Capítulo 8: SIFT (Scale Invariant Feature Transform)

https://buleria.unileon.es/bitstream/10612/11065/1/cap%208%20ConceptosyMetodosenVxC.pdf

[16] [17] [18] [19] SURF - Wikipedia, la enciclopedia libre

https://es.wikipedia.org/wiki/SURF

[20] [21] Transformada de Hough - Wikipedia, la enciclopedia libre

https://es.wikipedia.org/wiki/Transformada_de_Hough

[22] Conceptos clave de la detección de bordes para visión artificial

https://es.unitxlabs.com/key-concepts-edge-detection-machine-vision/

[23] [24] [25] Detector de bordes Canny, cómo contar objetos con OpenCV y Python

https://programarfacil.com/blog/vision-artificial/detector-de-bordes-canny-opencv/

[26] [27] [28] Aprendizaje supervisado frente a otros métodos en visión artificial

https://es.unitxlabs.com/supervised-learning-machine-vision-system-vs-other-methods/

[29] Detección de objetos - Wikipedia, la enciclopedia libre

https://es.wikipedia.org/wiki/Detecci%C3%B3n_de_objetos