Detección rápida de objetos mediante una cascada mejorada de Características

Pablo Viola
viola@merl.com
Laboratorios de investigación de Mitsubishi Electric
201 Broadway, 8.o FL
Cambridge, MA 02139

Michael Jones
mjones@crl.dec.com
Compaq CRL
Un centro de Cambridge
Cambridge, MA 02142

Resumen

Este documento describe un enfoque de aprendizaje automático para la detección de objetos visuales que es capaz de procesar imágenes extremadamente rápido y lograr altas tasas de detección. Este trabajo se distingue por tres contribuciones clave. La primera es la introducción de una nueva representación de imagen llamada "Imagen integral" que permite que las características utilizadas por nuestro detector se calculen muy rápidamente. El segundo es un algoritmo de aprendizaje, basado en AdaBoost, que selecciona un pequeño número de características visuales críticas de un conjunto más grande y produce clasificadores extremadamente eficientes [6]. La tercera contribución es un método para combinar clasificadores cada vez más complejos en una "cascada" que permite que las regiones de fondo de la imagen se descarten rápidamente mientras se gastan más cálculos en regiones prometedoras similares a objetos. La cascada puede verse como un mecanismo de foco de atención específico de un objeto que, a diferencia de los enfoques anteriores, proporciona garantías estadísticas de que es poco probable que las regiones descartadas contengan el objeto de interés. En el ámbito de la detección de rostros, el sistema produce tasas de detección comparables a las de los mejores sistemas anteriores. Utilizado en aplicaciones en tiempo real, el detector funciona a 15 fotogramas por segundo sin recurrir a la diferenciación de imágenes o la detección del color de la piel.

1. Introducción

Este documento reúne nuevos algoritmos y conocimientos para construir un marco para la detección de objetos robusta y extremadamente rápida. Este marco se demuestra y en parte está motivado por la tarea de detección de rostros. Con este fin, hemos construido un sistema de detección frontal de rostros que logra tasas de detección y falsos positivos equivalentes a los mejores resultados publicados [16, 12, 15, 11, 1]. Este sistema de detección de rostros se distingue más claramente de los enfoques anteriores en su capacidad para detectar rostros extremadamente rápido. Al operar en imágenes de 384 por 288 píxeles, las caras se tectó a 15 fotogramas por segundo en un Intel Pentium III convencional de 700 MHz. En otros sistemas de detección de rostros, se ha utilizado información auxiliar, como las diferencias de imagen en las secuencias de vídeo o el color de los píxeles en las imágenes en color, para lograr altas velocidades de cuadro. Nuestro sistema logra altas velocidades de cuadro trabajando solo con la información presente en una sola imagen en escala de grises. Estas fuentes alternativas de información también se pueden integrar con nuestro sistema para lograr velocidades de cuadro aún más altas.

Hay tres contribuciones principales de nuestro marco de detección de objetos. Presentaremos cada una de estas ideas brevemente a continuación y luego las describiremos en detalle en las secciones siguientes.

La primera contribución de este artículo es una nueva representación de imagen llamada *imagen integral* que permite una evaluación de funciones muy rápida. Motivado en parte por el trabajo de Papageorgiou et al. nuestro sistema de detección no trabaja directamente con intensidades de imagen [10]. Al igual que estos autores, usamos un conjunto de características que recuerdan a las funciones de Haar Basis (aunque también usaremos filtros relacionados que son más complejos que los filtros de Haar). Para calcular estas características muy rápidamente a muchas escalas, presentamos la representación de imagen integral para imágenes. La imagen integral se puede calcular a partir de una imagen utilizando algunas operaciones por píxel. Una vez calculada, cualquiera de estas características similares a Harr se puede calcular a cualquier escala o ubicación en *constante* hora.

La segunda contribución de este artículo es un método para construir un clasificador seleccionando un pequeño número de características importantes usando AdaBoost [6]. Dentro de cualquier subventana de imagen, el número total de características similares a Harr es muy grande, mucho mayor que el número de píxeles. Para garantizar una clasificación rápida, el proceso de aprendizaje debe excluir una gran mayoría de las funciones disponibles y centrarse en un pequeño conjunto de funciones críticas. Motivado por el trabajo de Tieu y Viola, la selección de características se logra a través de una simple modificación del procedimiento AdaBoost: el alumno débil está restringido de modo que cada clasificador débil devuelto puede depender solo de un

característica única [2]. Como resultado, cada etapa del proceso de refuerzo, que selecciona un nuevo clasificador débil, puede verse como un proceso de selección de características. AdaBoost proporciona un algoritmo de aprendizaje eficaz y límites sólidos en el rendimiento de la generalización [13, 9, 10].

La tercera gran contribución de este artículo es un método para combinar clasificadores sucesivamente más complejos en una estructura en cascada que aumenta drásticamente la velocidad del detector al centrar la atención en regiones prometedoras de la imagen. La noción detrás de los enfoques de enfoque de atención es que a menudo es posible determinar rápidamente en qué parte de una imagen puede aparecer un objeto [17, 8, 1]. El procesamiento más complejo está reservado solo para estas regiones prometedoras. La medida clave de tal enfoque es la tasa de "falsos negativos" del proceso de atención. Debe darse el caso de que todas, o casi todas, las instancias de objeto sean seleccionadas por el filtro atencional.

Describiremos un proceso para entrenar a un clasificador extremadamente simple y eficiente que puede usarse como un operador de foco de atención “supervisado”. El término supervisado se refiere al hecho de que el operador de atención está capacitado para detectar ejemplos de una clase en particular. En el dominio de la detección de rostros, es posible lograr menos del 1% de falsos negativos y del 40% de falsos positivos utilizando un clasificador construido a partir de dos características similares a las de Harr. El efecto de este filtro es reducir a más de la mitad el número de ubicaciones donde se debe evaluar el detector final.

Aquellas subventanas que no son rechazadas por el clasificador inicial son procesadas por una secuencia de clasificadores, cada uno ligeramente más complejo que el anterior. Si algún clasificador rechaza la subventana, no se realiza ningún procesamiento adicional. La estructura del proceso de detección en cascada es esencialmente la de un árbol de decisiones degenerado y, como tal, está relacionada con el trabajo de Geman y colegas [1, 4].

Un detector facial extremadamente rápido tendrá amplias aplicaciones prácticas. Estos incluyen interfaces de usuario, bases de datos de imágenes y teleconferencias. En aplicaciones donde no se necesitan velocidades de cuadro rápidas, nuestro sistema permitirá un posprocesamiento y análisis adicionales significativos. Además, nuestro sistema se puede implementar en una amplia gama de pequeños dispositivos de bajo consumo, incluidos procesadores portátiles y integrados. En nuestro laboratorio hemos implementado este detector facial en la computadora de mano Compaq iPaq y hemos logrado la detección a dos cuadros por segundo (este dispositivo tiene una potencia baja de 200 mips *Brazo fuerte* procesador que carece de hardware de punto flotante).

El resto del artículo describe nuestras contribuciones y una serie de resultados experimentales, incluida una descripción detallada de nuestra metodología experimental. La discusión de trabajos estrechamente relacionados se lleva a cabo al final de cada sección.

2. Características

Nuestro procedimiento de detección de objetos clasifica las imágenes basándose en el valor de características simples. Hay muchas motivaciones para usar funciones en lugar de los píxeles directamente. La razón más común es que las características pueden actuar para codificar el conocimiento de dominio ad-hoc que es difícil de aprender usando una cantidad finita de datos de entrenamiento. Para este sistema también existe una segunda motivación fundamental para las funciones: el sistema basado en funciones funciona mucho más rápido que un sistema basado en píxeles.

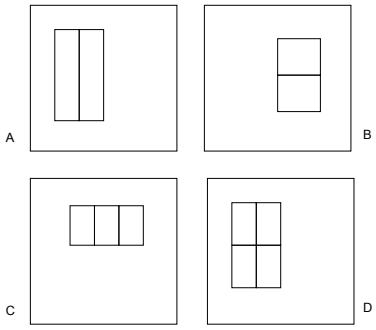

Figura 1: Ejemplo de características de rectángulo que se muestran en relación con la ventana de detección circundante. La suma de los píxeles que se encuentran dentro de los rectángulos blancos se resta de la suma de píxeles en los rectángulos grises. Las características de dos rectángulos se muestran en (A) y (B). La Figura (C) muestra una característica de tres rectángulos y (D) una característica de cuatro rectángulos.

Las características simples utilizadas recuerdan a las funciones básicas de Haar que han sido utilizadas por Papageorgiou et al. [10]. Más específicamente, utilizamos tres tipos de funciones. El valor de un *característica de dos rectángulos* es la diferencia entre la suma de los píxeles dentro de dos regiones rectangulares. Las regiones tienen el mismo tamaño y forma y son adyacentes horizontal o verticalmente (ver Figura 1). A *característica de tres rectángulos* calcula la suma dentro de dos rectángulos exteriores restados de la suma en un rectángulo central. Finalmente un *característica de cuatro rectángulos* calcula la diferencia entre pares diagonales de rectángulos.

Dado que la resolución base del detector es 24x24, el conjunto exhaustivo de características rectangulares es bastante grande, más de 180.000. Tenga en cuenta que, a diferencia de la base Haar, el conjunto de características del rectángulo está demasiado completo. [1].

2.1. Imagen integral

Las características de los rectángulos se pueden calcular muy rápidamente utilizando una representación intermedia de la imagen que llamamos imagen integral. [2] La imagen integral en el lugar contiene la suma de los píxeles arriba y a la izquierda de , inclusive:

[1] Una base completa no tiene dependencia lineal entre los elementos de la base y tiene el mismo número de elementos que el espacio de la imagen, en este caso 576. El conjunto completo de 180.000 mil características es muchas veces sobrecompleto.

[2] Existe una estrecha relación con las “tablas de áreas sumadas” que se utilizan en los gráficos [3]. Elegimos un nombre diferente aquí para enfatizar su uso para el análisis de imágenes, en lugar de para el mapeo de texturas.

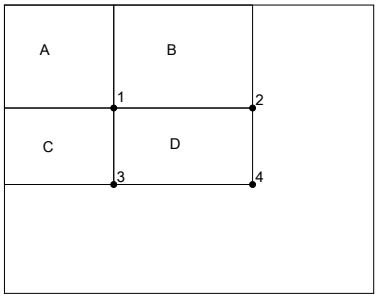

Figura 2: La suma de los píxeles dentro del rectángulo calculada con cuatro puede ser referencias de matriz. El valor de la imagen integral en la ubicación 1 es la suma de los píxeles en el rectángulo . El valor en la ubicación 2 es , en la ubicación 3 es , y en la ubicación 4 está . La suma dentro lata ser calculado como .

dónde es la imagen integral y es el origen imagen nal. Usando el siguiente par de recurrencias:

(1)

(2)

(dónde es la suma acumulada de las filas, , y) la imagen integral se puede calcular en una pasada sobre la imagen original.

Usando la imagen integral, cualquier suma rectangular se puede calcular en cuatro referencias de matriz (ver Figura 2). Claramente, la diferencia entre dos sumas rectangulares se puede calcular en ocho referencias. Dado que las características de dos rectángulos definidas anteriormente implican sumas rectangulares adyacentes, se pueden calcular en seis referencias de matriz, ocho en el caso de las características de tres rectángulos y nueve para las características de cuatro rectángulos.

2.2. Discusión de características

Las características rectangulares son algo primitivas en comparación con alternativas como los filtros orientables [5, 7]. Los filtros orientables, y sus parientes, son excelentes para el análisis detallado de límites, compresión de imágenes y análisis de texturas. Por el contrario, las características rectangulares, aunque sensibles a la presencia de bordes, barras y otras estructuras de imagen simples, son bastante toscas. A diferencia de los filtros orientables, las únicas orientaciones disponibles son vertical, horizontal y diagonal. Sin embargo, el conjunto de características rectangulares proporciona una rica representación de imágenes que respalda el aprendizaje efectivo. Junto con la imagen integral, la e fi ciencia del conjunto de características rectangulares proporciona una amplia compensación por su flexibilidad limitada.

3. Aprendizaje de funciones de clasificación

Dado un conjunto de funciones y un conjunto de entrenamiento de imágenes positivas y negativas, cualquier cantidad de enfoques de aprendizaje automático podría usarse para aprender una función de clasificación. En nuestro sistema se utiliza una variante de AdaBoost *ambos* para seleccionar un pequeño conjunto de funciones *y* capacitar al clasificador [6]. En su forma original, el algoritmo de aprendizaje AdaBoost se utiliza para mejorar el rendimiento de clasificación de un algoritmo de aprendizaje simple (a veces llamado débil). Hay una serie de garantías formales proporcionadas por el procedimiento de aprendizaje de AdaBoost. Freund y Schapire demostraron que el error de entrenamiento del clasificador fuerte se acerca a cero exponencialmente en el número de rondas. Más importante aún, se demostraron posteriormente una serie de resultados sobre el rendimiento de la generalización [14]. La idea clave es que el rendimiento de la generalización está relacionado con el margen de los ejemplos y que AdaBoost logra grandes márgenes rápidamente.

Recuerde que hay más de 180.000 características rectangulares asociadas con cada subventana de imagen, un número mucho mayor que el número de píxeles. Aunque cada característica se puede calcular de manera muy eficiente, calcular el conjunto completo es prohibitivamente caro. Nuestra hipótesis, que está confirmada por experimentos, es que un número muy pequeño de estas características puede combinarse para formar un clasificador eficaz. El principal desafío es encontrar estas características.

En apoyo de este objetivo, el algoritmo de aprendizaje débil está diseñado para seleccionar la característica de rectángulo único que mejor separa los ejemplos positivos y negativos (esto es similar al enfoque de [2] en el dominio de la recuperación de la base de datos de imágenes). Para cada característica, el alumno débil determina la función de clasificación de umbral óptima, de modo que el número mínimo de ejemplos se clasifica erróneamente. Un clasificador débil

por lo tanto, consta de una característica, un umbral y una paridad! indicando la dirección del signo de desigualdad:

si

de lo contrario

Aquí hay una subventana de 24x24 píxeles de una imagen. Consulte la Tabla 1 para obtener un resumen del proceso de refuerzo.

En la práctica, ninguna característica puede realizar la tarea de clasificación con un error bajo. Las características que se seleccionaron en las primeras rondas del proceso de refuerzo tuvieron tasas de error entre 0,1 y 0,3. Las características seleccionadas en rondas posteriores, a medida que la tarea se vuelve más difícil, arrojan tasas de error entre 0,4 y 0,5.

3.1. Discusión de aprendizaje

Se han propuesto muchos procedimientos de selección de características generales (consulte el capítulo 8 de [18] para una revisión). Nuestra aplicación final exigía un enfoque muy agresivo que descartara la gran mayoría de funciones. Para un problema de reconocimiento similar, Papageorgiou et al. propuso un esquema para la selección de características basado en la variación de características [10]. Demostraron buenos resultados al seleccionar 37 características de un total de 1734 características.

Roth y col. proponen un proceso de selección de características basado en la regla de aprendizaje del perceptrón exponencial de Winnow [11]. El proceso de aprendizaje de Winnow converge hacia una solución en la que muchos de estos pesos son cero. Sin embargo, una gran

Imágenes de ejemplo dadas dónde

para ejemplos negativos y positivos respectivamente.

Inicializar pesos por respectivamente, donde 'y (son el número de negativos y positivos respectivamente.

Para) :

1. Normalice los pesos,

así que eso es una distribución de probabilidad.

2. Para cada función, entrenar a un clasificador cuales está restringido al uso de una sola función. El error se evalúa con respecto a ,

3. Elija el clasi fi cador, , con el menor error .

4. Actualice los pesos:

dónde si ejemplo se clasi fi ca correctamente de lo contrario, y?

El clasificador final fuerte es:

oth. erwise

dónde

Tabla 1: El algoritmo AdaBoost para el aprendizaje de clasificadores. Cada ronda de impulso selecciona una característica de la 180.000 características potenciales.

se conservan varias características (quizás unos cientos o miles).

3.2. Resultados de aprendizaje

Si bien los detalles sobre la capacitación y el desempeño del sistema final se presentan en la Sección 5, varios resultados simples merecen ser discutidos. Los experimentos iniciales demostraron que un clasificador de cara frontal construido a partir de 200 características produce una tasa de detección del 95% con una tasa de falsos positivos de 1 en 14084. Estos resultados son convincentes, pero no suficientes para muchas tareas del mundo real. En términos de cálculo, este clasificador es probablemente más rápido que cualquier otro sistema publicado, y requiere 0,7 segundos para escanear una imagen de 384 por 288 píxeles. Desafortunadamente, la técnica más sencilla para mejorar el rendimiento de la detección, agregar características al clasificador, aumenta directamente el tiempo de cálculo.

Para la tarea de detección de rostros, las características iniciales del rectángulo seleccionadas por AdaBoost son significativas y fáciles de interpretar. La primera característica seleccionada parece centrarse en la propiedad de que la región de los ojos suele ser más oscura que la región de la nariz y las mejillas (ver Figura 3). Esta característica es relativamente grande en comparación con la subventana de detección y debería ser algo insensible al tamaño y la ubicación de la cara. La segunda característica seleccionada se basa en la propiedad de que los ojos son más oscuros que el puente de la nariz.

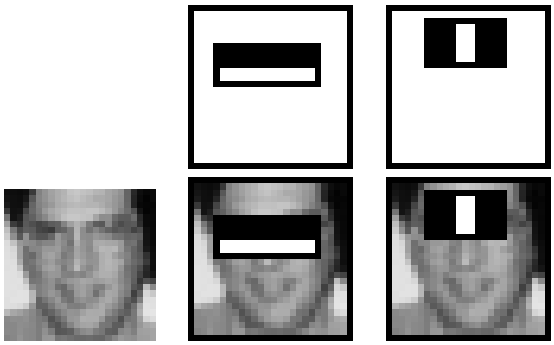

Figura 3: La primera y segunda características seleccionadas por AdaBoost. Las dos características se muestran en la fila superior y luego se superponen en una cara de entrenamiento típica en la fila inferior. La primera característica mide la diferencia de intensidad entre la región de los ojos y la región de la parte superior de las mejillas. La característica capitaliza la observación de que la región de los ojos suele ser más oscura que las mejillas. La segunda característica compara las intensidades en las regiones de los ojos con la intensidad en el puente de la nariz.

4. La cascada de la atención

Esta sección describe un algoritmo para construir una cascada de clasificadores que logra un mayor rendimiento de detección al tiempo que reduce radicalmente el tiempo de cálculo. La idea clave es que se pueden construir clasificadores potenciados más pequeños, y por lo tanto más eficientes, que rechacen muchas de las subventanas negativas mientras detectan casi todas las instancias positivas (es decir, el umbral de un clasificador potenciado puede ajustarse de modo que la tasa de falsos negativos sea cerca de cero). Los clasificadores más simples se utilizan para rechazar la mayoría de las subventanas antes de que se recurra a clasificadores más complejos para lograr tasas bajas de falsos positivos.

La forma general del proceso de detección es la de un árbol de decisiones degenerado, lo que llamamos una "cascada" (ver Figura 4). Un resultado positivo del primer clasificador desencadena la evaluación de un segundo clasificador que también se ha ajustado para lograr tasas de detección muy altas. Un resultado positivo del segundo clasificador activa un tercer clasificador, y así sucesivamente. Un resultado negativo en cualquier momento conduce al rechazo inmediato de la subventana.

Las etapas en la cascada se construyen capacitando a los clasificadores usando AdaBoost y luego ajustando el umbral para minimizar los falsos negativos. Tenga en cuenta que el umbral de AdaBoost predeterminado está diseñado para producir una tasa de error baja en los datos de entrenamiento. En general, un umbral más bajo produce una detección más alta.

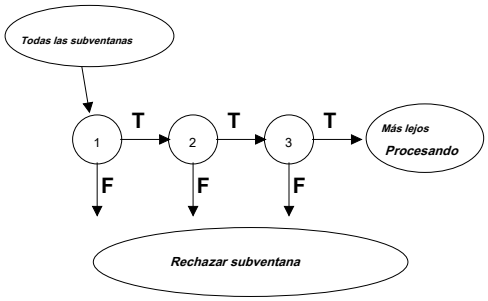

Figura 4: Representación esquemática de una cascada de detección. Se aplica una serie de clasificadores a cada subventana. El clasificador inicial elimina una gran cantidad de ejemplos negativos con muy poco procesamiento. Las capas posteriores eliminan negativos adicionales pero requieren cálculos adicionales. Después de varias etapas de procesamiento, el número de subventanas se ha reducido radicalmente. El procesamiento adicional puede tomar cualquier forma, como etapas adicionales de la cascada (como en nuestro sistema de detección) o un sistema de detección alternativo.

tasas de incidencia y tasas más altas de falsos positivos.

Por ejemplo, se puede construir un clasificador de primera etapa excelente a partir de un clasificador fuerte de dos características reduciendo el umbral para minimizar los falsos negativos. Comparado con un conjunto de entrenamiento de validación, el umbral se puede ajustar para detectar el 100% de las caras con una tasa de falsos positivos del 40%. Consulte la Figura 3 para obtener una descripción de las dos características utilizadas en este clasi fi cador.

El cálculo del clasificador de dos características equivale a unas 60 instrucciones de microprocesador. Parece difícil imaginar que cualquier filtro más simple pueda lograr tasas de rechazo más altas. En comparación, escanear una plantilla de imagen simple, o un perceptrón de una sola capa, requeriría al menos 20 veces más operaciones por subventana.

La estructura de la cascada refleja el hecho de que dentro de una sola imagen, la inmensa mayoría de las subventanas son negativas. Como tal, la cascada intenta rechazar tantos negativos como sea posible en la etapa más temprana posible. Si bien una instancia positiva desencadenará la evaluación de todos los clasificadores en la cascada, este es un evento extremadamente raro.

Al igual que un árbol de decisiones, los clasificadores posteriores se entrenan utilizando los ejemplos que pasan por todas las etapas anteriores. Como resultado, el segundo clasificador se enfrenta a una tarea más difícil que el primero. Los ejemplos que superan la primera etapa son "más difíciles" que los ejemplos típicos. Los ejemplos más difíciles que enfrentan los clasificadores más profundos empujan hacia abajo toda la curva de característica operativa del receptor (ROC). A una tasa de detección dada, los clasificadores más profundos tienen tasas de falsos positivos correspondientemente más altas.

4.1. Entrenando una cascada de clasificadores

El proceso de entrenamiento en cascada implica dos tipos de compensaciones. En la mayoría de los casos, los clasificadores con más características lograrán mayores tasas de detección y menores tasas de falsos positivos. Al mismo tiempo, los clasificadores con más características requieren más tiempo para calcular. En principio, se podría definir un marco de optimización en el que: i) el número de etapas del clasificador,

ii) el número de características en cada etapa, y iii) el umbral de cada etapa, se intercambian para minimizar el número esperado de características evaluadas. Desafortunadamente, encontrar este óptimo es un problema tremendamente difícil.

En la práctica, se utiliza un marco muy simple para producir un clasificador eficaz que es muy e fi ciente. Cada etapa de la cascada reduce la tasa de falsos positivos y disminuye la tasa de detección. Se selecciona un objetivo para la reducción mínima de falsos positivos y la disminución máxima de detección. Cada etapa se entrena agregando funciones hasta que se alcanzan las tasas de detección de objetivos y falsos positivos (estas tasas se determinan probando el detector en un conjunto de validación). Las etapas se agregan hasta que se alcanza el objetivo general de falsos positivos y la tasa de detección.

4.2. Discusión de la cascada del detector

La cascada completa de detección de rostros tiene 38 etapas con más de 6000 funciones. Sin embargo, la estructura en cascada da como resultado tiempos de detección promedio rápidos. En un conjunto de datos difícil, que contiene 507 caras y 75 millones de subventanas, las caras se detectan utilizando un promedio de 10 evaluaciones de características por subventana. En comparación, este sistema es aproximadamente 15 veces más rápido que una implementación del sistema de detección construido por Rowley et al. [3] [12]

Una noción similar a la cascada aparece en el sistema de detección de rostros descrito por Rowley et al. en el que se utilizan dos redes de detección [12]. Rowley y col. usó una red más rápida pero menos precisa para preseleccionar la imagen con el fin de encontrar regiones candidatas para una red más lenta y precisa. Aunque es difícil de determinar con exactitud, parece que el sistema de dos caras de red de Rowley et al. Es el detector de caras existente más rápido. [4]

La estructura del proceso de detección en cascada es esencialmente la de un árbol de decisiones degenerado y, como tal, está relacionada con el trabajo de Amit y Geman [1]. A diferencia de las técnicas que utilizan un detector fijo, Amit y Geman proponen un punto de vista alternativo en el que se utilizan co-ocurrencias inusuales de características de imagen simples para desencadenar la evaluación de un proceso de detección más complejo. De esta manera, no es necesario evaluar el proceso de detección completo en muchas de las posibles ubicaciones y escalas de imágenes. Si bien esta idea básica

[3] Henry Rowley amablemente nos proporcionó implementaciones de su sistema de detección para comparación directa. Los resultados reportados están en contra de su sistema más rápido. Es difícil de determinar a partir de la literatura publicada, pero el detector Rowley-Baluja-Kanade es ampliamente considerado el sistema de detección más rápido y ha sido probado en gran medida en problemas del mundo real.

[4] Otros detectores publicados se han olvidado de discutir el rendimiento en detalle o nunca han publicado tasas de detección y falsos positivos en un conjunto de entrenamiento grande y difícil.

es muy valioso, en su implementación es necesario evaluar primero algún detector de características en cada ubicación. Estas características luego se agrupan para encontrar co-ocurrencias inusuales. En la práctica, dado que la forma de nuestro detector y las características que utiliza son extremadamente eficientes, el costo amortizado de evaluar nuestro detector en *cada escala y ubicación* es mucho más rápido que buscar y agrupar bordes en toda la imagen.

En un trabajo reciente, Fleuret y Geman han presentado una técnica de detección de rostros que se basa en una "cadena" de pruebas para indicar la presencia de un rostro en una escala y ubicación particulares [4]. Las propiedades de la imagen medidas por Fleuret y Geman, disyunciones de bordes de escala fina, son bastante diferentes de las características rectangulares que son simples, existen en todas las escalas y son algo interpretables. Los dos enfoques también difieren radicalmente en su filosofía de aprendizaje. La motivación del proceso de aprendizaje de Fleuret y Geman es la estimación de densidad y la discriminación de densidad, mientras que nuestro detector es puramente discriminativo. Finalmente, la tasa de falsos positivos del enfoque de Fleuret y Geman parece ser más alta que la de enfoques anteriores como Rowley et al. y este enfoque. Lamentablemente, el documento no informa de resultados cuantitativos de este tipo.

Figura 5: Ejemplo de imágenes de la cara frontal vertical utilizadas para el entrenamiento.

5 resultados

Se entrenó un clasificador en cascada de 38 capas para detectar caras frontales verticales. Para entrenar el detector, se utilizó un conjunto de imágenes de entrenamiento faciales y no faciales. El conjunto de entrenamiento facial constaba de 4916 caras etiquetadas a mano escaladas y alineadas a una resolución base de 24 por 24 píxeles. Los rostros se extrajeron de imágenes descargadas durante un rastreo aleatorio de la red mundial. En la Figura 5 se muestran algunos ejemplos típicos de caras. Las subventanas sin caras utilizadas para entrenar el detector provienen de 9544 imágenes que se inspeccionaron manualmente y se encontró que no contenían ninguna cara. Hay alrededor de 350 millones de subventanas dentro de estas imágenes sin caras.

El número de características en las primeras cinco capas del detector es 1, 10, 25, 25 y 50 características, respectivamente. Las capas restantes tienen cada vez más características. El número total de entidades en todas las capas es 6061.

Cada clasificador en la cascada se entrenó con 4916 caras de entrenamiento (más sus imágenes de espejo vertical para un total de 9832 caras de entrenamiento) y 10,000 subventanas sin rostro (también de tamaño 24 por 24 píxeles) usando el procedimiento de entrenamiento Adaboost. Para el clasificador de una característica inicial, los ejemplos de entrenamiento sin rostro se recopilaron seleccionando subventanas aleatorias de un conjunto de 9544 imágenes que no contenían rostros. Los ejemplos sin rostros utilizados para entrenar capas posteriores se obtuvieron escaneando la cascada parcial a través de las imágenes sin rostros y recolectando falsos positivos. Se recopiló un máximo de 10000 de estas subventanas no faciales para cada capa.

Velocidad del detector final

La velocidad del detector en cascada está directamente relacionada con el número de características evaluadas por subventana escaneada. Evaluado en el conjunto de prueba MIT + CMU [12], se evalúa un promedio de 10 características de un total de 6061 por subventana. Esto es posible porque una gran mayoría de subventanas son rechazadas por la primera o segunda capa de la cascada. En un procesador Pentium III de 700 Mhz, el detector facial puede procesar una imagen de 384 por 288 píxeles en aproximadamente 0,067 segundos (utilizando una escala inicial de 1,25 y un tamaño de paso de 1,5 que se describe a continuación). Esto es aproximadamente 15 veces más rápido que el detector RowleyBaluja-Kanade [12] y aproximadamente 600 veces más rápido que el detector Schneiderman-Kanade [15].

Procesamiento de imágenes

Todas las subventanas de ejemplo utilizadas para el entrenamiento se normalizaron la varianza para minimizar el efecto de las diferentes condiciones de iluminación. Por lo tanto, la normalización también es necesaria durante la detección. La varianza de una subventana de imagen se puede calcular rápidamente utilizando un par de imágenes integrales. Recordar que

— , donde esta el estandarte

desviación, es la media y es el valor de píxel dentro

la subventana. La media de una subventana se puede calcular utilizando la imagen integral. La suma de píxeles cuadrados se calcula usando una imagen integral de la imagen cuadrada (es decir, se usan dos imágenes integrales en el proceso de escaneo). Durante el escaneo, el efecto de normalización de la imagen se puede lograr multiplicando posteriormente los valores de las características en lugar de multiplicar previamente los píxeles.

Escaneando el detector

El detector final se escanea a través de la imagen en múltiples escalas y ubicaciones. El escalado se logra escalando el detector mismo, en lugar de escalar la imagen. Este proceso tiene sentido porque las características se pueden evaluar en cualquier

Detecciones falsas / Detector	10	31	50	sesenta y cinco	78	95	167
Viola-Jones	76,1%	88,4%	91,4%	92,0%	92,1%	92,9%	93,9%
Viola-Jones (votando)	81,1%	89,7%	92,1%	93,1%	93,1%	93,2%	93,7%
Rowley-Baluja-Kanade	83,2%	86,0%	-	-	-	89,2%	90,1%
Schneiderman-Kanade	-	-	-	94,4%	-	-	-
Roth-Yang-Ahuja	-	-	-	-	(94,8%)	-	-

Tabla 2: Tasas de detección para varios números de falsos positivos en el conjunto de prueba MIT + CMU que contiene 130 imágenes y 507 caras.

escala con el mismo costo. Se obtuvieron buenos resultados utilizando un conjunto de escalas separadas por un factor de 1,25.

El detector también se escanea a través de la ubicación. Las ubicaciones posteriores se obtienen desplazando la ventana algunos píxeles. Este proceso de cambio se ve afectado por la escala del detector: si la escala actual es, la ventana se desplaza por

, donde es la operación de redondeo. La elección de afecta tanto la velocidad del detector como así como precisión. Los resultados que presentamos son para . Podemos lograr una aceleración significativa configurando con solo una ligera disminución en la precisión.

Integración de múltiples detecciones

Dado que el detector final es insensible a pequeños cambios en la traducción y la escala, generalmente se producirán múltiples detecciones alrededor de cada cara en una imagen escaneada. Lo mismo ocurre a menudo con algunos tipos de falsos positivos. En la práctica, a menudo tiene sentido devolver una detección final por rostro. Con este fin, es útil posprocesar las subventanas detectadas para combinar detecciones superpuestas en una sola detección.

En estos experimentos, las detecciones se combinan de una manera muy sencilla. El conjunto de detecciones se divide primero en subconjuntos disjuntos. Dos detecciones están en el mismo subconjunto si sus regiones limítrofes se superponen. Cada partición produce una única detección final. Las esquinas de la región límite final son el promedio de las esquinas de todas las detecciones del conjunto.

Experimentos en un conjunto de pruebas del mundo real

Probamos nuestro sistema en el equipo de prueba de cara frontal MIT + CMU [12]. Este conjunto consta de 130 imágenes con 507 caras frontales etiquetadas. La curva AROC que muestra el rendimiento de nuestro detector en este equipo de prueba se muestra en la Figura 6. Para crear la curva ROC, el umbral del clasificador de capa final se ajusta de a . Ajustando el umbral a producirá una tasa de detección de 0.0 y una tasa de falsos positivos de 0.0. Ajustando el umbral a , sin embargo, aumenta tanto la tasa de detección como la tasa de falsos positivos, pero solo hasta cierto punto. Ninguna tasa puede ser mayor que la tasa de la cascada de detección menos la capa final. En efecto, un umbral de es equivalente a quitar esa capa. Más lejos aumentar las tasas de detección y de falsos positivos requiere disminuir el umbral del siguiente clasificador en la cascada.

Por lo tanto, para construir una curva ROC completa, se eliminan las capas del clasificador. Usamos el *número* de falsos positivos en contraposición a la *Velocidad* de falsos positivos para el eje x de la curva ROC para facilitar la comparación con otros sistemas. Para calcular la tasa de falsos positivos, simplemente divida por el número total de subventanas escaneadas. En nuestros experimentos, el número de subventanas escaneadas es 75 081 800.

Desafortunadamente, la mayoría de los resultados publicados anteriormente sobre la detección de rostros solo han incluido un régimen operativo único (es decir, un solo punto en la curva ROC). Para facilitar la comparación con nuestro detector, hemos enumerado nuestra tasa de detección de las tasas de falsos positivos informadas por los otros sistemas. La Tabla 2 enumera la tasa de detección para varios números de detecciones falsas para nuestro sistema, así como para otros sistemas publicados. Para los resultados de Rowley-Baluja-Kanade [12], se probaron varias versiones diferentes de su detector que arrojaron una serie de resultados diferentes en los que se enumeran todos bajo el mismo título. Para el detector Roth-Yang-Ahuja [11], informaron su resultado en el conjunto de prueba MIT + CMU menos 5 imágenes que contenían caras dibujadas con líneas eliminadas.

La Figura 7 muestra la salida de nuestro detector facial en algunas imágenes de prueba del equipo de prueba MIT + CMU.

Un esquema de votación simple para mejorar aún más los resultados

En la tabla 2 también mostramos los resultados de ejecutar tres detectores (el de 38 capas uno descrito anteriormente más dos detectores entrenados de manera similar) y emitir el voto mayoritario de los tres detectores. Esto mejora la tasa de detección y elimina más falsos positivos. La mejora sería mayor si los detectores fueran más independientes. La correlación de sus errores da como resultado una modesta mejora con respecto al mejor detector individual.

6. Conclusiones

Hemos presentado un enfoque para la detección de objetos que minimiza el tiempo de cálculo al tiempo que logra una alta precisión de detección. El enfoque se utilizó para construir un sistema de detección de rostros que es aproximadamente 15 veces más rápido que cualquier enfoque anterior.

Este artículo reúne nuevos algoritmos, representaciones y conocimientos que son bastante genéricos y bien

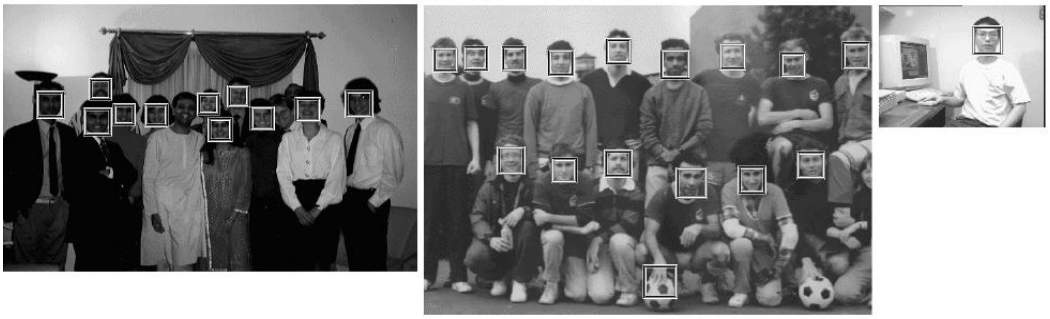

Figura 7: Salida de nuestro detector facial en una serie de imágenes de prueba del equipo de prueba MIT + CMU.

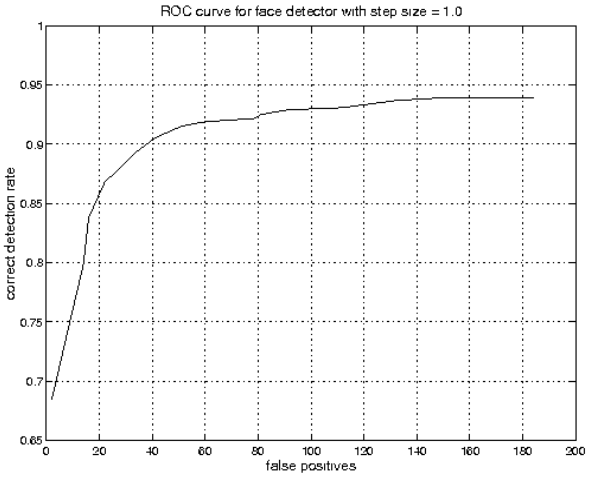

Figura 6: Curva ROC para nuestro detector facial en el equipo de prueba MIT + CMU. El detector se ejecutó usando un tamaño de paso de 1.0 y una escala inicial de 1.0 (75.081.800 subventanas escaneadas).

tienen una aplicación más amplia en la visión por computadora y el procesamiento de imágenes.

Por último, este artículo presenta un conjunto de experimentos detallados sobre un conjunto de datos de detección de rostros difíciles que ha sido ampliamente estudiado. Este conjunto de datos incluye rostros en una amplia gama de condiciones que incluyen: iluminación, escala, pose y variación de la cámara. Los experimentos con un conjunto de datos tan grande y complejo son difíciles y requieren mucho tiempo. No obstante, es poco probable que los sistemas que funcionan en estas condiciones sean frágiles o limitados a un único conjunto de condiciones. Más importante aún, es poco probable que las conclusiones extraídas de este conjunto de datos sean artefactos experimentales.

Referencias

[1] Y. Amit, D. Geman y K. Wilder. Inducción de forma conjunta características y clasificadores de árboles, 1997.

[2] Anónimo. Anónimo. En *Anónimo,* 2000.

[3] F. Cuervo. Tablas de área sumada para mapeo de texturas. En *Actas de SIGGRAPH,* volumen 18 (3), páginas 207–212, 1984.

[4] F. Fleuret y D. Geman. Detección de rostros de gruesa a fina. *En t. J. Visión por computadora,* 2001.

[5] William T. Freeman y Edward H. Adelson. El diseño y uso de filtros orientables. *Transacciones IEEE sobre análisis de patrones e inteligencia de máquinas,* 13 (9): 891–906, 1991.

[6] Yoav Freund y Robert E. Schapire. Una decisión teórica generalización del aprendizaje en línea y una aplicación al impulso. En *Teoría del aprendizaje computacional: Eurocolt '95,* páginas 23–37. Springer-Verlag, 1995.

[7] H. Greenspan, S. Belongie, R. Gooodman, P. Perona, S. Rakmierda, y C. Anderson. Filtros piramidales orientables sobrecompletos e invariancia de rotación. En *Actas de la Conferencia IEEE sobre Visión por Computador y Reconocimiento de Patrones,* 1994.

[8] L. Itti, C. Koch y E. Niebur. Un modelo basado en la prominencia atención visual para un análisis rápido de la escena. *IEEE Patt. Anal. Mach. Intell.,* 20 (11): 1254-1259, noviembre de 1998.

[9] Edgar Osuna, Robert Freund y Federico Girosi. Capacitación support vector machines: una aplicación para la detección de rostros. En *Actas de la Conferencia IEEE sobre Visión por Computador y Reconocimiento de Patrones,* 1997.

[10] C. Papageorgiou, M. Oren y T. Poggio. Un marco general trabajar para la detección de objetos. En *Congreso Internacional de Visión por Computador,* 1998.

[11] D. Roth, M. Yang y N. Ahuja. Un detector facial a base de nieve. En *Procesamiento de información neuronal 12,* 2000.

[12] H. Rowley, S. Baluja y T. Kanade. Basado en redes neuronales Detección de rostro. En *IEEE Patt. Anal. Mach. Intell.,* volumen 20, páginas 22–38, 1998.

[13] RE Schapire, Y. Freund, P. Bartlett y WS Lee. Aumentaring the margin: una nueva explicación de la eficacia de los métodos de votación. *Ana. Stat.,* 26 (5): 1651-1686, 1998.

[14] Robert E. Schapire, Yoav Freund, Peter Bartlett y Wee Sun Lee. Aumento del margen: una nueva explicación de la eficacia de los métodos de votación. En *Actas de la Decimocuarta Conferencia Internacional sobre Aprendizaje Automático,* 1997.

[15] H. Schneiderman y T. Kanade. Un método estadístico para 3D detección de objetos aplicada a rostros y coches. En *Congreso Internacional de Visión por Computador,* 2000.

[16] K. Sung y T. Poggio. Aprendizaje basado en ejemplos para visualización
detección de rostros basada. En *IEEE Patt. Anal. Mach. Intell.,* volumen 20, páginas 39–51, 1998.

[17] JK Tsotsos, SM Culhane, WYK Wai, YH Lai, N. Davis,
y F. Nu fl o. Modelado de la atención visual a través de la sintonización selectiva. *Revista de inteligencia artificial,* 78 (1-2): 507–545, octubre de 1995.

[18] Andrew Webb. *Reconocimiento estadístico de patrones.* Universidad de Oxford
Versity Press, Nueva York, 1999.