

**ARQUITECTURAS DE FILTRO EN BUCLE BASADAS EN NN DE BAJA
COMPLEJIDAD CON CONVOLUCIÓN SEPARABLE**

Esta solicitud reivindica prioridad a la solicitud
5 de patente de los EE. UU. N.º 18/738.842, presentada el 10
de junio de 2024 y la solicitud de patente provisional de
los EE. UU. N.º 63/507.651, presentada el 12 de junio de
2023, cuyos contenidos completos se incorporan como
referencia. La solicitud de patente de los EE. UU. núm.
10 18/738.842, presentada el 10 de junio de 2024, reivindica
el beneficio de la solicitud de patente provisional de los
EE. UU. 63/507.651, presentada el 12 de junio de 2023.

CAMPO TÉCNICO

15

Esta divulgación se relaciona con la
codificación de video y la decodificación de video.

ANTECEDENTES

20

Las capacidades de video digital pueden
incorporarse en una amplia gama de dispositivos, incluidos
televisores digitales, sistemas de transmisión directa
digital, sistemas de transmisión inalámbrica, asistentes
25 digitales personales (PDA, por sus siglas en inglés),
computadores portátiles o de escritorio, tabletas,

lectores de libros electrónicos, cámaras digitales, dispositivos de grabación digital, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, teléfonos de radio celulares o satelitales, los llamados “teléfonos inteligentes”, dispositivos de teleconferencia de video, dispositivos de transmisión de video, y lo similar. Los dispositivos de video digital implementan técnicas de codificación de video, tales como las descritas en los estándares definidos por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Parte 10, Codificación de video avanzada (AVC, por sus siglas en inglés), ITU-T H.265/Codificación de video de alta eficiencia (HEVC, por sus siglas en inglés), ITU-T H.266/Codificación de video versátil (VVC, por sus siglas en inglés), y extensiones de tales estándares, así como códecs/formatos de video de propiedad exclusiva, tales como AOMedia Video 1 (AV1, por sus siglas en inglés) que fuera desarrollado por Alliance for Open Media. Los dispositivos de video pueden transmitir, recibir, codificar, decodificar y/o almacenar información de video digital de manera más eficiente implementando tales técnicas de codificación de video.

Las técnicas de codificación de video incluyen predicción espacial (intraimagen) y/o predicción temporal (interimagen) para reducir o eliminar la redundancia inherente de las secuencias de video. Para la codificación de video basada en bloques, una porción de video (p. ej.,

una imagen de video o una parte de una imagen de video) puede dividirse en bloques de video, que también pueden denominarse unidades de árbol de codificación (CTU, por sus siglas en inglés), unidades de codificación (CU, por sus siglas en inglés) y/o nodos de codificación. Los bloques de video en una porción intracodificada (I) de una imagen se codifican usando predicción espacial con respecto a muestras de referencia en bloques vecinos en la misma imagen. Los bloques de video en una porción intercodificada (P o B) de una imagen pueden usar predicción espacial con respecto a muestras de referencia en bloques vecinos en la misma imagen o predicción temporal con respecto a muestras de referencia en otras imágenes de referencia. Las imágenes pueden denominarse cuadros, y las imágenes de referencia pueden denominarse cuadros de referencia.

SUMARIO

En general, esta divulgación describe técnicas para el filtrado en bucle basado en red neuronal (NN, por sus siglas en inglés) para codificación de video. Más particularmente, esta divulgación describe ejemplos de un filtro en bucle basado en NN simplificado. Se ha desarrollado una Arquitectura Unificada (UA, por sus siglas en inglés) para filtros en bucle basados en NN. Sin embargo, la UA consume mucha energía y procesamiento y, por lo tanto, no es

apropiada para el uso en dispositivos con limitaciones de energía y/o procesamiento, tales como dispositivos móviles alimentados por batería. Las técnicas de esta divulgación proporcionan un filtro en bucle basado en NN operable en tal
5 dispositivo restringido.

En un ejemplo, un método incluye: recibir una imagen de los datos de video; reconstruir la imagen de los datos de video; y aplicar un filtro basado en red neuronal (NN) a la imagen reconstruida de los datos de video, en
10 donde el filtro basado en NN comprende un filtro unificado que comprende un bloque de cabecera, un bloque de transición, uno o más bloques troncales y un bloque de cola, y en donde al menos uno del bloque de cabecera, el bloque de transición, el uno o más bloques troncales, o el
15 bloque de cola comprende una descomposición poliádica canónica (CP) con convolución separable.

En otro ejemplo, un dispositivo incluye una o más memorias configuradas para almacenar los datos de video; y uno o más procesadores, uno o más procesadores configurados
20 para: recibir una imagen de datos de video; reconstruir la imagen de datos de video; y aplicar un filtro basado en red neuronal (NN) a la imagen reconstruida de datos de video, en donde el filtro basado en NN comprende un filtro unificado que comprende un bloque de cabecera, un bloque
25 de transición, uno o más bloques troncales y un bloque de cola, y en donde al menos uno del bloque de cabecera, el

bloque de transición, el uno o más bloques troncales, o el bloque de cola comprende una descomposición poliádica canónica (CP) con convolución separable.

En otro ejemplo, un dispositivo incluye medios para
5 recibir una imagen de datos de video; medios para reconstruir la imagen de datos de video; y medios para aplicar un filtro basado en red neuronal (NN) a la imagen reconstruida de datos de video, en donde el filtro basado en NN comprende un filtro unificado que comprende un bloque de cabecera, un bloque de
10 transición, uno o más bloques troncales y un bloque de cola, y en donde al menos uno del bloque de cabecera, el bloque de transición, el uno o más bloques troncales, o el bloque de cola comprende una descomposición poliádica canónica (CP) con convolución separable.

En otro ejemplo, los medios de almacenamiento
15 legibles por computador se codifican con instrucciones que, cuando se ejecutan, hacen que uno o más procesadores programables: reciban una imagen de datos de video; reconstruir la imagen de datos de video; y aplicar un filtro
20 basado en red neuronal (NN) a la imagen reconstruida de datos de video, en donde el filtro basado en NN comprende un filtro unificado que comprende un bloque de cabecera, un bloque de transición, uno o más bloques troncales y un bloque de cola, y en donde al menos uno del bloque de
25 cabecera, el bloque de transición, el uno o más bloques

troncales, o el bloque de cola comprende una descomposición poliádica canónica (CP) con convolución separable.

Los detalles de uno o más ejemplos se exponen en las figuras adjuntas y en la siguiente descripción. Otras
5 características, objetos y ventajas resultarán evidentes a partir de la descripción, los dibujos y las reivindicaciones.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

10 La Figura 1 es un diagrama en bloque que ilustra un ejemplo de sistema de codificación y decodificación de video que puede realizar las técnicas de esta divulgación.

La Figura 2 es un diagrama en bloque que ilustra un marco de codificación de video híbrido de ejemplo.

15 La Figura 3 ilustra un ejemplo de estructura de predicción jerárquica con un tamaño de grupo de imágenes (GOP, por sus siglas en inglés) igual a 16.

La Figura 4 es un diagrama en bloque que ilustra un ejemplo de filtro basado en una red neuronal de
20 convolución (CNN, por sus siglas en inglés) con cuatro capas.

La Figura 5 es un diagrama en bloque que ilustra un filtro basado en CNN de ejemplo con muestras de entrada acojinadas y datos suplementarios.

La Figura 6 es un diagrama en bloque que ilustra otro filtro basado en CNN de ejemplo con muestras de entrada acojinadas y datos suplementarios.

La Figura 7 es un diagrama conceptual que ilustra un bloque residual de atención de la Figura 6.

La Figura 8 es un diagrama en bloque que ilustra otro filtro basado en CNN de ejemplo con muestras de entrada acojinadas y datos suplementarios.

La Figura 9 es un diagrama en bloque que ilustra un ejemplo de una arquitectura de filtro basada en CNN simplificada con muestras de entrada acojinadas y datos suplementarios.

La Figura 10 es un diagrama conceptual que ilustra una estructura de bloques residuales de ejemplo de la Figura 9.

La Figura 11 es un diagrama en bloque que ilustra otro ejemplo de una arquitectura de filtro basada en CNN simplificada con muestras de entrada acojinadas y datos suplementarios.

La Figura 12 es un diagrama conceptual que ilustra una estructura de bloque de filtro de ejemplo de la Figura 11.

La Figura 13 es un diagrama en bloque que ilustra una arquitectura de descomposición de CNN de ejemplo.

La Figura 14 es un diagrama en bloque que ilustra una red troncal de extracción de características multiescala de ejemplo con convolución de dos componentes.

La Figura 15 es un diagrama en bloque que ilustra
5 un filtro unificado de ejemplo con un modelo unido (luma y croma unidos).

La Figura 16 es un diagrama en bloque que ilustra un bloque troncal de ejemplo.

La Figura 17 es un diagrama en bloque que ilustra
10 un filtro unificado de ejemplo con modelos separados de luma y croma (luma).

La Figura 18 es un diagrama en bloque que ilustra un filtro unificado de ejemplo con modelos separados de luma y croma (croma).

15 La Figura 19 es un diagrama en bloque que ilustra una arquitectura de red de filtro en bucle de baja complejidad de ejemplo de acuerdo con uno o más aspectos de esta divulgación.

La Figura 20 es un diagrama en bloque que ilustra
20 un bloque troncal de ejemplo de acuerdo con uno o más aspectos de esta divulgación.

La Figura 21 es un diagrama en bloque que ilustra una descomposición CP de ejemplo, de acuerdo con uno o más aspectos de esta divulgación.

La Figura 22 es un diagrama de flujo que ilustra técnicas de filtrado en bucle basadas en NN de ejemplo de acuerdo con uno o más aspectos de esta divulgación.

La Figura 23 es un diagrama en bloque que ilustra
5 un codificador de video de ejemplo que puede realizar las técnicas de esta divulgación.

La Figura 24 es un diagrama en bloque que ilustra un decodificador de video de ejemplo que puede realizar las técnicas de esta divulgación.

10 La Figura 25 es un diagrama de flujo que ilustra un ejemplo de método para codificar un bloque actual de acuerdo con las técnicas de esta divulgación.

La Figura 26 es un diagrama de flujo que ilustra un ejemplo de método para decodificar un bloque actual de
15 acuerdo con las técnicas de esta divulgación.

DESCRIPCIÓN DETALLADA

La codificación de video (p. ej., codificación de video y/o decodificación de video) implica típicamente predecir un bloque de datos de video a partir de un bloque ya codificado de datos de video en la misma imagen (p. ej., intrapredicción) o un bloque ya codificado de datos de video en una imagen diferente (p. ej., interpredicción). En algunos casos, el codificador de video también calcula los datos residuales al comparar el bloque de predicción con el bloque original. Por lo tanto, los datos residuales representan una diferencia entre el bloque de predicción y el bloque original. Para reducir el número de bits necesarios para señalar los datos residuales, el codificador de video transforma y cuantifica los datos residuales y señala los datos residuales transformados y cuantificados en el flujo de bits codificado. La compresión lograda por los procesos de transformación y cuantificación puede tener pérdidas, lo que significa que los procesos de transformación y cuantificación pueden introducir distorsión en los datos de video decodificados.

Un decodificador de video decodifica y añade los datos residuales al bloque de predicción para producir un bloque de video reconstruido que coincide más estrechamente con el bloque de video original que el bloque de predicción solo. Debido a la pérdida introducida por la transformación

y cuantificación de los datos residuales, el bloque reconstruido puede tener distorsión o artefactos. Un tipo común de artefacto o distorsión se conoce como "blockiness" (fragmentación en bloques), donde los límites de los bloques
5 usados para codificar los datos de video son visibles.

Para mejorar aún más la calidad del video decodificado, un decodificador de video puede realizar una o más operaciones de filtrado en los bloques de video reconstruidos. Ejemplos de estas operaciones de filtrado
10 incluyen la filtración de desbloqueo, el filtrado de desplazamiento adaptativo de muestras (SAO, por sus siglas en inglés) y el filtrado de bucle adaptativo (ALF, por sus siglas en inglés). Los parámetros para estas operaciones de filtrado pueden determinarse mediante un codificador de video
15 y señalizarse explícitamente en el flujo de bits de video codificado o pueden determinarse implícitamente mediante un decodificador de video sin necesidad de que los parámetros se señalicen explícitamente en el flujo de bits de video codificado.

20 Esta divulgación describe arquitecturas de filtrado en bucle (ILF, por sus siglas en inglés) basadas en red neuronal (NN) para fines de codificación de video. Las técnicas de esta divulgación pueden mejorar el rendimiento de codificación bajo limitaciones de complejidad y
25 requerimientos de memoria. En algunos ejemplos, las arquitecturas de esta divulgación pueden estar basadas en

red neuronal de convolución (CNN). En algunos ILF de CNN, la reducción de la complejidad puede lograrse por descomposición (p. ej., utilizando convoluciones separables) en lugar de convoluciones multidimensionales convencionales.

5 En general, esta divulgación describe técnicas para el filtrado en bucle basado en NN para codificación de video. Los codificadores de video (p. ej., codificadores y/o decodificadores) pueden tener limitaciones de complejidad y requerimiento de memoria, tales como cuando tales
10 codificadores de video se implementan en un dispositivo móvil. Las técnicas divulgadas pueden mejorar el rendimiento de codificación bajo limitaciones de complejidad y requerimiento de memoria, lo que da como resultado una reproducción de video de mayor calidad que las técnicas
15 existentes bajo tales restricciones. La reducción de la complejidad para la arquitectura se puede lograr al utilizar una descomposición poliádica canónica (CP) con convolución separable en lugar de una convolución convencional con núcleos multidimensionales, eliminando o simplificando capas
20 de activación y/o eliminando o moviendo rama(s) multiescala. Las técnicas descritas en esta divulgación están relacionadas con el filtrado en bucle basado en NN, sin embargo, son aplicables a cualquier herramienta de codificación de video basada en NN que consuma datos de entrada con ciertas
25 propiedades estadísticas. Las técnicas pueden usarse en el contexto de códecs de video avanzados, tales como extensiones

de VVC o la próxima generación de estándares de codificación de video, y cualquier otro códec de video.

La Figura 1 es un diagrama en bloque que ilustra un ejemplo de sistema de codificación y decodificación de video 100 que puede realizar las técnicas de esta divulgación. Las técnicas de esta divulgación se dirigen generalmente a codificar (codificar y/o decodificar) datos de video. En general, los datos de video incluyen cualquier dato para procesar un video. Por lo tanto, los datos de video pueden incluir video sin codificar, video codificado, video decodificado (p. ej., reconstruido) y metadatos de video, tales como datos de señalización.

Como se muestra en la Figura 1, el sistema 100 incluye un dispositivo fuente 102 que proporciona datos de video codificados a ser decodificados y visualizados por un dispositivo de destino 116, en este ejemplo. En particular, el dispositivo fuente 102 proporciona los datos de video al dispositivo de destino 116 a través de un medio legible por computador 110. El dispositivo fuente 102 y el dispositivo de destino 116 pueden ser o incluir cualquiera de una amplia gama de dispositivos, tales como computadores de escritorio, computadores portátiles (es decir, portátiles), dispositivos móviles, tabletas, decodificadores, terminales telefónicos tales como teléfonos inteligentes, televisores, cámaras, dispositivos de visualización, reproductores de medios digitales,

consolas de videojuegos, dispositivo de transmisión de video, dispositivos receptores de difusión, o lo similar. En algunos casos, el dispositivo fuente 102 y el dispositivo de destino 116 pueden estar equipados para la comunicación inalámbrica y, por lo tanto, pueden denominarse dispositivos de comunicación inalámbrica.

En el ejemplo de la Figura 1, el dispositivo fuente 102 incluye una fuente de video 104, una memoria 106, un codificador de video 200 y una interfaz de salida 108. El dispositivo de destino 116 incluye una interfaz de entrada 122, un decodificador de video 300, una memoria 120 y un dispositivo de visualización 118. De acuerdo con esta divulgación, el codificador de video 200 del dispositivo fuente 102 y el decodificador de video 300 del dispositivo de destino 116 pueden configurarse para aplicar las técnicas para filtrado en bucle basado en NN. Por lo tanto, el dispositivo fuente 102 representa un ejemplo de un dispositivo de codificación de video, mientras que el dispositivo de destino 116 representa un ejemplo de un dispositivo de decodificación de video. En otros ejemplos, un dispositivo de origen y un dispositivo de destino pueden incluir otros componentes o disposiciones. Por ejemplo, el dispositivo fuente 102 puede recibir datos de video desde una fuente de video externa, tal como una cámara externa. De la misma manera, el dispositivo de destino 116 puede

interactuar con un dispositivo de visualización externo, en lugar de incluir un dispositivo de visualización integrado.

El sistema 100 como se muestra en la Figura 1 es simplemente un ejemplo. En general, cualquier dispositivo de codificación y/o decodificación de video digital puede realizar técnicas para filtrado en bucle basado en NN. El dispositivo fuente 102 y el dispositivo de destino 116 son simplemente ejemplos de tales dispositivos de codificación en los que el dispositivo fuente 102 genera datos de video codificados para su transmisión al dispositivo de destino 116. Esta divulgación se refiere a un dispositivo de “codificación” como un dispositivo que realiza la codificación (codificación y/o decodificación) de datos. Por lo tanto, el codificador de video 200 y el decodificador de video 300 representan ejemplos de dispositivos de codificación, en particular, un codificador de video y un decodificador de video, respectivamente. En algunos ejemplos, el dispositivo fuente 102 y el dispositivo de destino 116 pueden operar de una manera sustancialmente simétrica de tal manera que cada uno del dispositivo fuente 102 y el dispositivo de destino 116 incluye componentes de codificación y decodificación de video. Por lo tanto, el sistema 100 puede soportar transmisión de video unidireccional o bidireccional entre el dispositivo fuente 102 y el dispositivo de destino 116, p. ej., para

transmisión de video, reproducción de video, difusión de video o videotelefonía.

En general, la fuente de video 104 representa una fuente de datos de video (es decir, datos de video sin
5 codificar) y proporciona una serie secuencial de imágenes (también denominados “cuadros”) de los datos de video al codificador de video 200, que codifica datos para las imágenes. La fuente de video 104 del dispositivo fuente 102 puede incluir un dispositivo de captura de video, tal como
10 una cámara de video, un archivo de video que contiene video sin procesar previamente capturado, y/o una interfaz de alimentación de video para recibir video de un proveedor de contenido de video. Como alternativa adicional, la fuente de video 104 puede generar datos basados en gráficos
15 informáticos como el video fuente, o una combinación de video en directo, video archivado y video generado por computador. En cada caso, el codificador de video 200 codifica los datos de video capturados, precapturados o generados por computador. El codificador de video 200 puede
20 reorganizar las imágenes del orden recibido (a veces denominado “orden de visualización”) en un orden de codificación para la codificación. El codificador de video 200 puede generar un flujo de bits que incluye datos de video codificados. El dispositivo fuente 102 puede entonces
25 emitir los datos de video codificados a través de la interfaz de salida 108 en un medio legible por computador

110 para recepción y/o recuperación por, p. ej., la interfaz de entrada 122 del dispositivo de destino 116.

La memoria 106 del dispositivo fuente 102 y la memoria 120 del dispositivo de destino 116 representan memorias de propósito general. En algunos ejemplos, las memorias 106, 120 pueden almacenar datos de video sin procesar, p. ej., video sin procesar procedente de la fuente de video 104 y datos de video decodificados sin procesar del decodificador de video 300. Adicional o alternativamente, las memorias 106, 120 pueden almacenar instrucciones de software ejecutables por, p. ej., el codificador de video 200 y el decodificador de video 300, respectivamente. Aunque la memoria 106 y la memoria 120 se muestran por separado del codificador de video 200 y del decodificador de video 300 en este ejemplo, debe entenderse que el codificador de video 200 y el decodificador de video 300 también pueden incluir memorias internas para fines funcionalmente similares o equivalentes. Además, las memorias 106, 120 pueden almacenar datos de video codificados, p. ej., salida desde el codificador de video 200 y entrada al decodificador de video 300. En algunos ejemplos, porciones de las memorias 106, 120 pueden asignarse como uno o más búferes de video, p. ej., para almacenar datos de video sin procesar, decodificados y/o codificados.

El medio legible por computador 110 puede representar cualquier tipo de medio o dispositivo capaz de

transportar los datos de video codificados desde el dispositivo fuente 102 al dispositivo de destino 116. En un ejemplo, el medio legible por computador 110 representa un medio de comunicación para permitir que el dispositivo fuente

5 102 transmita datos de video codificados directamente al dispositivo de destino 116 en tiempo real, p. ej., a través de una red de radiofrecuencia o una red basada en computador. La interfaz de salida 108 puede modular una señal de transmisión que incluye los datos de video codificados, y la

10 interfaz de entrada 122 puede demodular la señal de transmisión recibida, de acuerdo con un estándar de comunicación, tal como un protocolo de comunicación inalámbrica. El medio de comunicación puede incluir cualquier medio de comunicación inalámbrico o por cable, tal como un

15 espectro de radiofrecuencia (RF) o una o más líneas de transmisión física. El medio de comunicación puede formar parte de una red basada en paquetes, tal como una red de área local, una red de área amplia o una red global como Internet. El medio de comunicación puede incluir enrutadores,

20 conmutadores, estaciones base o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo fuente 102 al dispositivo de destino 116.

En algunos ejemplos, el dispositivo fuente 102 puede emitir datos codificados desde la interfaz de salida

25 108 al dispositivo de almacenamiento 112. De manera similar, el dispositivo de destino 116 puede acceder a datos

codificados desde el dispositivo de almacenamiento 112 a través de la interfaz de entrada 122. El dispositivo de almacenamiento 112 puede incluir cualquiera de una variedad de medios de almacenamiento de datos distribuidos o a los que se accede localmente, tales como un disco duro, discos Blu-ray, DVD, CD-ROM, memoria flash, memoria volátil o no volátil, o cualquier otro medio de almacenamiento digital adecuado para almacenar datos de video codificados.

En algunos ejemplos, el dispositivo fuente 102 puede emitir datos de video codificados al servidor de archivos 114 u otro dispositivo de almacenamiento intermedio que puede almacenar los datos de video codificados generados por el dispositivo fuente 102. El dispositivo de destino 116 puede acceder a datos de video almacenados desde el servidor de archivos 114 a través de transmisión o descarga.

El servidor de archivos 114 puede ser cualquier tipo de dispositivo servidor capaz de almacenar datos de video codificados y transmitir esos datos de video codificados al dispositivo de destino 116. El servidor de archivos 114 puede representar un servidor web (p. ej., para un sitio web), un servidor configurado para proporcionar un servicio de protocolo de transferencia de archivos (tal como protocolo de transferencia de archivos (FTP, por sus siglas en inglés) o protocolo de entrega de archivos a través de transporte unidireccional (FLUTE, por sus siglas en inglés), un dispositivo de red de entrega de

contenido (CDN, por sus siglas en inglés), un servidor de protocolo de transferencia de hipertexto (HTTP, por sus siglas en inglés), un servicio de difusión multidifusión multimedia (MBMS, por sus siglas en inglés), y/o un
5 dispositivo de almacenamiento conectado a la red (NAS, por sus siglas en inglés). Adicional o alternativamente, el servidor de archivos 114 puede implementar uno o más protocolos de transmisión HTTP, tales como transmisión dinámica adaptativa sobre HTTP (DASH, por sus siglas en
10 inglés), transmisión en vivo HTTP (HLS, por sus siglas en inglés), protocolo de transmisión en tiempo real (RTSP, por sus siglas en inglés), transmisión dinámica HTTP, o lo similar.

El dispositivo de destino 116 puede acceder a
15 datos de video codificados desde el servidor de archivos 114 a través de cualquier conexión de datos estándar, incluida una conexión a Internet. Esto puede incluir un canal inalámbrico (p. ej., una conexión Wi-Fi), una conexión por cable (p. ej., línea de abonado digital (DSL,
20 por sus siglas en inglés)), módem por cable, etc.), o una combinación de ambos que sea adecuada para acceder a datos de video codificados almacenados en el servidor de archivos 114. La interfaz de entrada 122 puede configurarse para funcionar de acuerdo con uno o más de los diversos
25 protocolos discutidos anteriormente para recuperar o recibir datos de medios desde el servidor de archivos 114,

u otros protocolos de este tipo para recuperar datos de medios.

La interfaz de salida 108 y la interfaz de entrada 122 pueden representar transmisores/receptores inalámbricos, módems, componentes de red cableados (p. ej., tarjetas Ethernet), componentes de comunicación inalámbrica que operan de acuerdo con cualquiera de una variedad de estándares IEEE 802.11 u otros componentes físicos. En ejemplos donde la interfaz de salida 108 y la interfaz de entrada 122 incluyen componentes inalámbricos, la interfaz de salida 108 y la interfaz de entrada 122 pueden configurarse para transferir datos, tales como datos de video codificados, de acuerdo con un estándar de comunicación celular, tal como 4G, 4G-LTE (Long-Term Evolution), LTE Advanced, 5G, o lo similar. En algunos ejemplos donde la interfaz de salida 108 incluye un transmisor inalámbrico, la interfaz de salida 108 y la interfaz de entrada 122 pueden configurarse para transferir datos, tales como datos de video codificados, de acuerdo con otros estándares inalámbricos, tales como una especificación IEEE 802.11, una especificación IEEE 802.15 (p. ej., ZigBee™), un estándar Bluetooth™, o lo similar. En algunos ejemplos, el dispositivo fuente 102 y/o el dispositivo de destino 116 pueden incluir dispositivos respectivos de sistema en un chip (SoC, por sus siglas en inglés). Por ejemplo, el dispositivo fuente 102 puede incluir un dispositivo SoC para realizar la funcionalidad atribuida al

codificador de video 200 y/o a la interfaz de salida 108, y el dispositivo de destino 116 puede incluir un dispositivo SoC para realizar la funcionalidad atribuida al decodificador de video 300 y/o a la interfaz de entrada 122.

5 Las técnicas de esta divulgación pueden aplicarse a la codificación de video en soporte de cualquiera de una variedad de aplicaciones multimedia, tales como transmisiones de televisión por aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, 10 transmisiones de video en flujo continuo por Internet, tales como transmisión dinámica adaptativa sobre HTTP (DASH), video digital que se codifica en un medio de almacenamiento de datos, decodificación de video digital almacenado en un medio de almacenamiento de datos u otras aplicaciones.

15 La interfaz de entrada 122 del dispositivo de destino 116 recibe un flujo de bits de video codificado desde un medio legible por computador 110 (p. ej., un medio de comunicación, un dispositivo de almacenamiento 112, un servidor de archivos 114 o lo similar). El flujo de bits de 20 video codificado puede incluir información de señalización definida por el codificador de video 200, que también es usado por el decodificador de video 300, tal como elementos de sintaxis que tienen valores que describen características y/o procesamiento de bloques de video u otras unidades 25 codificadas (p. ej., porciones, imágenes, grupos de imágenes, secuencias, o lo similar). El dispositivo de visualización

118 muestra imágenes decodificadas de los datos de video decodificados a un usuario. El dispositivo de visualización 118 puede representar cualquiera de una variedad de dispositivos de visualización tales como una pantalla de cristal líquido (LCD, por sus siglas en inglés), una pantalla de plasma, una pantalla de diodo emisor de luz orgánico (OLED, por sus siglas en inglés) u otro tipo de dispositivo de visualización.

Aunque no se muestra en la Figura 1, en algunos ejemplos, el codificador de video 200 y el decodificador de video 300 pueden estar integrados cada uno con un codificador de audio y/o decodificador de audio (p. ej., códec de audio), y pueden incluir unidades MUX-DEMUX apropiadas, u otro hardware y/o software, para manejar flujos multiplexados que incluyen tanto audio como video en un flujo de datos común. Los códecs de audio de ejemplo pueden incluir AAC, AC-3, AC-4, ALAC, ALS, AMBE, AMR, AMR-WB (G.722.2), AMR-WB+, aptx (varias versiones), ATRAC, BroadVoice (BV16, BV32), CELT, AC-3 (E-AC-3) mejorado, EVS, 20 FLAC, G.711, G.722, G.722.1, G.722.2 (AMR-WB). G.723.1, G.726, G.728, G.729, G.729.1, GSM-FR, HE-AAC, iLBC, iSAC, LA Lyra, Monkey's Audio, MP1, MP2 (MPEG-1, 2 Audio Layer II), MP3, Musepack, Nellymoser Asao, OptimFROG, Opus, Sac, 25 TwinVQ, USAC, Vorbis (Ogg), WavPack, y Windows Media Aud.

El codificador de video 200 y el decodificador de video 300 pueden implementarse cada uno como cualquiera de una variedad de circuitos de codificador y/o decodificador adecuados, tales como uno o más microprocesadores, procesadores de señal digital (DSP, por sus siglas en inglés), circuitos integrados de aplicación específica (ASIC, por sus siglas en inglés), matrices de puertas programables en campo (FPGA, por sus siglas en inglés), lógica discreta, software, hardware, firmware o cualquier combinación de estos. Cuando las técnicas se implementan parcialmente en software, un dispositivo puede almacenar instrucciones para el software en un medio legible por computador no transitorio adecuado, y ejecutar las instrucciones en hardware usando uno o más procesadores para realizar las técnicas de esta divulgación. Cada uno del codificador de video 200 y del decodificador de video 300 puede incluirse en uno o más codificadores o decodificadores, cualquiera de los cuales puede estar integrado como parte de un codificador/decodificador combinado (CODEC, por sus siglas en inglés) en un dispositivo respectivo. Un dispositivo que incluye el codificador de video 200 y/o el decodificador de video 300 puede implementar el codificador de video 200 y/o el decodificador de video 300 en circuitos de procesamiento tal como un circuito integrado y/o un microprocesador. Tal dispositivo puede ser un dispositivo de comunicación

inalámbrica, tal como un teléfono celular, o cualquier otro tipo de dispositivo descrito en la presente.

El codificador de video 200 y el decodificador de video 300 pueden funcionar de acuerdo con un estándar de codificación de video, tal como ITU-T H.265, también denominado Codificación de video de alta eficiencia (HEVC) o extensiones de este, tales como las extensiones de codificación de video multivista y/o escalable. Alternativamente, el codificador de video 200 y el decodificador de video 300 pueden funcionar de acuerdo con otros estándares propietarios o de la industria, tales como ITU-T H.266, también denominado codificación de video versátil (VVC). En otros ejemplos, el codificador de video 200 y el decodificador de video 300 pueden funcionar de acuerdo con un códec/formato de video propietario, tal como AOMedia Video 1 (AV1, por sus siglas en inglés), extensiones de AV1 y/o versiones sucesores de AV1 (p. ej., AV2). En otros ejemplos, el codificador de video 200 y el decodificador de video 300 pueden funcionar de acuerdo con otros formatos propietarios o estándares de la industria. Sin embargo, las técnicas de esta divulgación no se limitan a ningún estándar o formato de codificación particular. En general, el codificador de video 200 y el decodificador de video 300 pueden configurarse para realizar las técnicas de esta divulgación junto con cualquier técnica de

codificación de video que usen redes neuronales, tales como la codificación de video basada en NN.

En general, el codificador de video 200 y el decodificador de video 300 pueden realizar codificación basada en bloques de imágenes. El término “bloque” generalmente se refiere a una estructura que incluye datos a ser procesados (p. ej., codificados, decodificados o usados de cualquier otra manera en el proceso de codificación y/o decodificación). Por ejemplo, un bloque puede incluir una matriz bidimensional de muestras de luminancia y/o datos de crominancia. En general, el codificador de video 200 y el decodificador de video 300 pueden codificar datos de video representados en un formato YUV (p. ej., Y, Cb, Cr). Es decir, en lugar de codificar datos rojos, verdes y azules (RGB, por sus siglas en inglés) para muestras de una imagen, el codificador de video 200 y el decodificador de video 300 pueden codificar componentes de luminancia y crominancia, donde los componentes de crominancia pueden incluir componentes de crominancia tanto de tono rojo como de tono azul. En algunos ejemplos, el codificador de video 200 convierte los datos con formato RGB recibidos en una representación YUV antes de la codificación, y el decodificador de video 300 convierte la representación YUV al formato RGB. Alternativamente, las unidades de procesamiento previo y

posterior (no mostradas) pueden realizar esta conversiones.

Esta divulgación puede referirse generalmente a la codificación (p. ej., codificación y decodificación) de
5 imágenes para incluir el proceso de codificación o decodificación de datos de la imagen. De manera similar, esta divulgación puede referirse a la codificación de bloques de una imagen para incluir el proceso de codificación o decodificación de datos para los bloques,
10 p. ej., predicción y/o codificación residual. Un flujo de bits de video codificado generalmente incluye una serie de valores para elementos de sintaxis representativos de decisiones de codificación (p. ej., modos de codificación) y partición de imágenes en bloques. Por lo tanto, las
15 referencias a la codificación de una imagen o un bloque deben entenderse generalmente como valores de codificación para elementos de sintaxis que forman la imagen o bloque.

HEVC define varios bloques, incluidas unidades de codificación (CU), unidades de predicción (PU, por sus
20 siglas en inglés) y unidades de transformación (TU, por sus siglas en inglés). De acuerdo con HEVC, un codificador de video (como el codificador de video 200) divide una unidad de árbol de codificación (CTU) en CU de acuerdo con una estructura de árbol cuaternario. Es decir, el
25 codificador de video divide las CTU y las CU en cuatro cuadrados iguales, no superpuestos, y cada nodo del árbol

cuaternario tiene cero o cuatro nodos secundarios. Los nodos sin nodos secundarios pueden denominarse “nodos hoja”, y las CU de dichos nodos hoja pueden incluir una o más PU y/o una o más TU. El codificador de video puede
5 además particionar las PU y las TU. Por ejemplo, en HEVC, un árbol cuaternario residual (RQT, por sus siglas en inglés) representa la partición de las TU. En HEVC, las PU representan datos de interpredicción, mientras que las TU representan datos residuales. Las CU que son
10 intrapredichas incluyen información de intrapredicción, como una indicación intramodo.

Como otro ejemplo, el codificador de video 200 y el decodificador de video 300 pueden configurarse para funcionar de acuerdo con VVC. De acuerdo con VVC, un
15 codificador de video (tal como el codificador de video 200) divide una imagen en una pluralidad de CTU. El codificador de video 200 puede particionar una CTU de acuerdo con una estructura de árbol, tal como una estructura de árbol cuaternario-binario (QTBT, por sus siglas en inglés) o una
20 estructura de árbol multitipo (MTT, por sus siglas en inglés). La estructura QTBT elimina los conceptos de múltiples tipos de partición, como la separación entre CU, PU y TU de HEVC. Una estructura QTBT incluye dos niveles: un primer nivel particionado de acuerdo con la partición de
25 árbol cuaternario, y un segundo nivel particionado de acuerdo con la partición de árbol binario. Un nodo raíz de la

estructura QTBT corresponde a una CTU. Los nodos hoja de los árboles binarios corresponden a CU.

En una estructura de partición MTT, los bloques pueden particionarse usando una partición de árbol cuaternario (QT, por sus siglas en inglés), una partición de árbol binario (BT, por sus siglas en inglés) y uno o más tipos de particiones de árbol triple (TT, por sus siglas en inglés) (también llamado árbol ternario (TT)). Una partición de árbol triple o ternario es una partición donde un bloque se divide en tres subbloques. En algunos ejemplos, una partición de árbol triple o ternario divide un bloque en tres subbloques sin dividir el bloque original a través del centro. Los tipos de partición en MTT (p. ej., QT, BT y TT) pueden ser simétricos o asimétricos.

15 Cuando se opera de acuerdo con el códec AV1, el codificador de video 200 y el decodificador de video 300 pueden configurarse para codificar datos de video en bloques. En AV1, el bloque de codificación más grande que se puede procesar se llama superbloque. En AV1, un superbloque puede ser muestras de 128x128 luma o muestras de 64x64 luma. Sin embargo, en los formatos de codificación de video sucesores (p. ej., AV2), un superbloque puede definirse por diferentes tamaños de muestra de luma (p. ej., más grandes). En algunos ejemplos, un superbloque es el nivel superior de un árbol cuaternario de bloques. El codificador de video 200 puede dividir además un superbloque en bloques de codificación más

pequeños. El codificador de video 200 puede dividir un superbloque y otros bloques de codificación en bloques más pequeños usando particiones cuadradas o no cuadradas. Los bloques no cuadrados pueden incluir bloques $N/2 \times N$, $N \times N/2$,
5 $N/4 \times N$ y $N \times N/4$. El codificador de video 200 y el decodificador de video 300 pueden realizar procesos separados de predicción y transformación en cada uno de los bloques de codificación.

AV1 también define un mosaico de datos de video. Un mosaico es una matriz rectangular de superbloques que
10 pueden codificarse independientemente de otros mosaicos. Es decir, el codificador de video 200 y el decodificador de video 300 pueden codificar y decodificar, respectivamente, bloques de codificación dentro de un mosaico sin usar datos de video de otros mosaicos. Sin embargo, el codificador de
15 video 200 y el decodificador de video 300 pueden realizar el filtrado a través de los límites del mosaico. Los mosaicos pueden ser uniformes o no uniformes en tamaño. La codificación basada en mosaico puede permitir el procesamiento paralelo y/o subprocesos múltiples para
20 implementaciones de codificador y decodificador.

En algunos ejemplos, el codificador de video 200 y el decodificador de video 300 pueden usar una única estructura QTBT o MTT para representar cada uno de los componentes de luminancia y crominancia, mientras que en
25 otros ejemplos, el codificador de video 200 y el decodificador de video 300 pueden usar dos o más estructuras

QTBT o MTT, tales como una estructura QTBT/MTT para el componente de luminancia y otra estructura QTBT/MTT para ambos componentes de crominancia (o dos estructuras QTBT/MTT para los respectivos componentes de crominancia).

5 El codificador de video 200 y el decodificador de video 300 pueden configurarse para usar particiones de árbol cuaternario, partición QTBT, partición MTT, partición de superbloque u otras estructuras de partición.

 En algunos ejemplos, una CTU incluye un bloque de
10 árbol de codificación (CTB) de muestras de luma, dos CTB correspondientes de muestras de croma de una imagen que tiene tres matrices de muestras, o un CTB de muestras de una imagen monocromática o una imagen que se codifica usando tres planos de color separados y estructuras de sintaxis usadas para
15 codificar las muestras. Un CTB puede ser un bloque $N \times N$ de muestras para algún valor de N tal que la división de un componente en CTB es una partición. Un componente es una matriz o muestra única de una de las tres matrices (luma y dos croma) que componen una imagen en formato de color 4:2:0,
20 4:2:2, o 4:4:4 o la matriz o una sola muestra de la matriz que compone una imagen en formato monocromático. En algunos ejemplos, un bloque de codificación es un bloque $M \times N$ de muestras para algunos valores de M y N tal que una división de un CTB en bloques de codificación es una partición.

25 Los bloques (p. ej., CTU o CU) pueden agruparse de diversas maneras en una imagen. Como ejemplo, un bloque

puede referirse a una región rectangular de filas CTU dentro de un mosaico particular en una imagen. Un mosaico puede ser una región rectangular de CTU dentro de una columna de mosaico particular y una fila de mosaico particular en una imagen. Una columna de mosaico se refiere a una región rectangular de CTU que tiene una altura igual a la altura de la imagen y un ancho especificado por elementos de sintaxis (p. ej., en un conjunto de parámetros de imagen). Una fila de mosaico se refiere a una región rectangular de CTU que tiene una altura especificada por elementos de sintaxis (p. ej., en un conjunto de parámetros de imagen) y un ancho igual al ancho de la imagen.

En algunos ejemplos, un mosaico puede dividirse en múltiples bloques, cada uno de los cuales puede incluir una o más filas de CTU dentro del mosaico. Un mosaico que no está particionado en múltiples bloques también puede denominarse bloque. Sin embargo, un bloque que es un verdadero subconjunto de un mosaico no puede denominarse mosaico. Los bloques en una imagen también pueden estar dispuestos en una porción. Una porción puede ser un número entero de bloques de una imagen que puede estar contenido exclusivamente en una única unidad de capa de abstracción de red (NAL, por sus siglas en inglés). En algunos ejemplos, una porción incluye un número de mosaicos completos o solo una secuencia consecutiva de bloques completos de un mosaico.

Esta divulgación puede usar “NxN” y “N por N” indistintamente para referirse a las dimensiones de muestra de un bloque (tal como una CU u otro bloque de video) en términos de dimensiones verticales y horizontales, p. ej.,
5 muestras 16x16 o muestras 16 por 16. En general, una CU 16x16 tendrá 16 muestras en una dirección vertical ($y = 16$) y 16 muestras en una dirección horizontal ($x = 16$). De la misma manera, una CU NxN generalmente tiene N muestras en una dirección vertical y N muestras en una dirección
10 horizontal, donde N representa un valor entero no negativo. Las muestras en una CU pueden estar dispuestas en filas y columnas. Además, las CU no necesitan tener necesariamente el mismo número de muestras en la dirección horizontal que en la dirección vertical. Por ejemplo, las CU pueden incluir
15 muestras de NxM, donde M no es necesariamente igual a N.

El codificador de video 200 codifica datos de video para las CU que representan predicción y/o información residual, y otra información. La información de predicción indica cómo se predecirá la CU para formar
20 un bloque de predicción para la CU. La información residual generalmente representa las diferencias muestra por muestra entre las muestras de la CU antes de la codificación y el bloque de predicción.

Para predecir una CU, el codificador de video 200
25 generalmente puede formar un bloque de predicción para la CU a través de la interpredicción o intrapredicción. La

interpredicción generalmente se refiere a predecir la CU a partir de datos de una imagen previamente codificada, mientras que la intrapredicción generalmente se refiere a predecir la CU a partir de datos previamente codificados de la misma imagen. Para realizar la interpredicción, el codificador de video 200 puede generar el bloque de predicción usando uno o más vectores de movimiento. El codificador de video 200 puede realizar generalmente una búsqueda de movimiento para identificar un bloque de referencia que coincide estrechamente con la CU, p. ej., en términos de diferencias entre la CU y el bloque de referencia. El codificador de video 200 puede calcular una métrica de diferencia usando una suma de diferencia absoluta (SAD, por sus siglas en inglés), suma de diferencias cuadradas (SSD, por sus siglas en inglés), diferencia absoluta media (MAD, por sus siglas en inglés), diferencias cuadradas medias (MSD, por sus siglas en inglés) u otros cálculos de diferencia similares para determinar si un bloque de referencia coincide estrechamente con la CU actual. En algunos ejemplos, el codificador de video 200 puede predecir la CU actual usando predicción unidireccional o predicción bidireccional.

Algunos ejemplos de VVC también proporcionan un modo de compensación de movimiento afín, que puede considerarse un modo de interpredicción. En el modo de compensación de movimiento afín, el codificador de video 200

puede determinar dos o más vectores de movimiento que representan movimiento no traslacional, como acercamiento o alejamiento, rotación, movimiento de perspectiva u otros tipos de movimiento irregular.

5 Para realizar la intrapredicción, el codificador de video 200 puede seleccionar un modo de intrapredicción para generar el bloque de predicción. Algunos ejemplos de VVC proporcionan sesenta y siete modos de intrapredicción, incluidos varios modos direccionales, así como el modo
10 plano y el modo DC. En general, el codificador de video 200 selecciona un modo de intrapredicción que describe muestras vecinas a un bloque actual (p. ej., un bloque de una CU) a partir del cual predecir muestras del bloque actual. Dichas muestras generalmente pueden estar arriba,
15 arriba y a la izquierda, o a la izquierda del bloque actual en la misma imagen que el bloque actual, suponiendo que el codificador de video 200 codifica CTU y CU en orden de escaneo ráster (de izquierda a derecha, de arriba a abajo).

El codificador de video 200 codifica datos que
20 representan el modo de predicción para un bloque actual. Por ejemplo, para los modos de interpredicción, el codificador de video 200 puede codificar datos que representan cuál de los diversos modos de interpredicción disponibles se usa, así como información de movimiento para
25 el modo correspondiente. Para la interpredicción unidireccional o bidireccional, por ejemplo, el codificador

de video 200 puede codificar vectores de movimiento usando predicción de vectores de movimiento avanzada (AMVP, por sus siglas en inglés) o modo de fusión. El codificador de video 200 puede usar modos similares para codificar
5 vectores de movimiento para el modo de compensación de movimiento afín.

AV1 incluye dos técnicas generales para codificar y decodificar un bloque de codificación de datos de video. Las dos técnicas generales son la intrapredicción (p. ej.,
10 predicción intracuadro o predicción espacial) y la interpredicción (p. ej., predicción intercuadro o predicción temporal). En el contexto de AV1, cuando se predicen bloques de un cuadro actual de datos de video usando un modo de intrapredicción, el codificador de video 200 y el
15 decodificador de video 300 no usan datos de video de otros cuadros de datos de video. Para la mayoría de los modos de intrapredicción, el codificador de video 200 codifica bloques de un cuadro actual basado en la diferencia entre los valores de muestra en el bloque actual y los valores
20 predichos generados a partir de muestras de referencia en el mismo cuadro. El codificador de video 200 determina los valores predichos generados a partir de las muestras de referencia basado en el modo de intrapredicción.

Después de la predicción, tal como la
25 intrapredicción o la interpredicción de un bloque, el codificador de video 200 puede calcular datos residuales para

el bloque. El dato residual, como un bloque residual, representa las diferencias muestra por muestra entre el bloque y un bloque de predicción para el bloque, formado usando el modo de predicción correspondiente. El codificador de video 200 puede aplicar una o más transformaciones al bloque residual, para producir datos transformados en un dominio de transformación en lugar del dominio de muestra. Por ejemplo, el codificador de video 200 puede aplicar una transformación de coseno discreta (DCT), una transformación entera, una transformación de ondícula o una transformación conceptualmente similar a datos de video residuales. Adicionalmente, el codificador de video 200 puede aplicar una transformación secundaria después de la primera transformación, tal como una transformación secundaria no separable dependiente del modo (MDNSST, por sus siglas en inglés), una transformación dependiente de señal, una transformación de Karhunen-Loeve (KLT, por sus siglas en inglés), o lo similar. El codificador de video 200 produce coeficientes de transformación después de la aplicación de una o más transformaciones.

Como se señaló anteriormente, después de cualquier transformación para producir coeficientes de transformación, el codificador de video 200 puede realizar la cuantificación de los coeficientes de transformación. La cuantificación generalmente se refiere a un proceso en el que los coeficientes de transformación se cuantifican para

posiblemente reducir la cantidad de datos usados para representar los coeficientes de transformación, proporcionando una compresión adicional. Al realizar el proceso de cuantificación, el codificador de video 200 puede

5 reducir la profundidad de bits asociada con algunos o todos los coeficientes de transformación. Por ejemplo, el codificador de video 200 puede redondear un valor de n bit a un valor de m bit durante la cuantificación, donde n es mayor que m . En algunos ejemplos, para realizar la cuantificación,

10 el codificador de video 200 puede realizar un desplazamiento a la derecha bit a la derecha del valor a cuantificar.

Después de la cuantificación, el codificador de video 200 puede escanear los coeficientes de transformación, al producir un vector unidimensional a

15 partir de la matriz bidimensional que incluye los coeficientes de transformación cuantificados. El escaneo puede estar diseñado para colocar coeficientes de transformación de mayor energía (y por lo tanto de menor frecuencia) en la parte frontal del vector y para colocar

20 coeficientes de transformación de energía más baja (y por lo tanto de mayor frecuencia) en la parte posterior del vector. En algunos ejemplos, el codificador de video 200 puede usar un orden de exploración predefinido para escanear los coeficientes de transformación cuantificados

25 para producir un vector serializado, y después codificar por entropía los coeficientes de transformación

cuantificados del vector. En otros ejemplos, el codificador de video 200 puede realizar un escaneo adaptativo. Después de escanear los coeficientes de transformación cuantificados para formar el vector unidimensional, el
5 codificador de video 200 puede codificar por entropía el vector unidimensional, p. ej., de acuerdo con la codificación aritmética binaria adaptativa al contexto (CABAC). El codificador de video 200 también puede codificar por entropía valores para elementos de sintaxis
10 que describen metadatos asociados con los datos de video codificados para su uso por el decodificador de video 300 en la decodificación de los datos de video.

Para realizar CABAC, el codificador de video 200 puede asignar un contexto dentro de un modelo de contexto
15 a un símbolo a transmitir. El contexto puede relacionarse, por ejemplo, con si los valores vecinos del símbolo son de valor cero o no. La determinación de probabilidad puede basarse en un contexto asignado al símbolo.

El codificador de video 200 puede generar además
20 datos de sintaxis, tales como datos de sintaxis basados en bloques, datos de sintaxis basados en imágenes y datos de sintaxis basados en secuencia, para el decodificador de video 300, p. ej., en un encabezado de imagen, un encabezado de bloque, un encabezado de porción u otros datos de
25 sintaxis, tales como un conjunto de parámetros de secuencia (SPS, por sus siglas en inglés), un conjunto de parámetros

de imagen (PPS, por sus siglas en inglés) o un conjunto de parámetros de video (VPS, por sus siglas en inglés). El decodificador de video 300 también puede decodificar tales datos de sintaxis para determinar cómo decodificar los
5 datos de video correspondientes.

De esta manera, el codificador de video 200 puede generar un flujo de bits que incluye datos de video codificados, p. ej., elementos de sintaxis que describen la partición de una imagen en bloques (p. ej., CU) y predicción
10 y/o información residual para los bloques. En última instancia, el decodificador de video 300 puede recibir el flujo de bits y decodificar los datos de video codificados.

En general, el decodificador de video 300 realiza un proceso recíproco al realizado por el codificador de
15 video 200 para decodificar los datos de video codificados del flujo de bits. Por ejemplo, el decodificador de video 300 puede decodificar valores para elementos de sintaxis del flujo de bits usando CABAC de una manera sustancialmente similar, aunque recíproca a, el proceso de
20 codificación CABAC del codificador de video 200. Los elementos de sintaxis pueden definir información de partición para la partición de una imagen en CTU, y la partición de cada CTU de acuerdo con una estructura de partición correspondiente, tal como una estructura QTBT,
25 para definir las CU de la CTU. Los elementos de sintaxis

pueden definir adicionalmente predicción e información residual para bloques (p. ej., CU) de datos de video.

La información residual puede estar representada por, por ejemplo, coeficientes de transformación cuantificados. El decodificador de video 300 puede cuantizar de manera inversa y transformar de manera inversa los coeficientes de transformación cuantificados de un bloque para reproducir un bloque residual para el bloque. El decodificador de video 300 usa un modo de predicción señalado (intra o interpredicción) e información de predicción relacionada (p. ej., información de movimiento para la interpredicción) para formar un bloque de predicción para el bloque. El decodificador de video 300 puede entonces combinar el bloque de predicción y el bloque residual (muestra por muestra) para reproducir el bloque original. El decodificador de video 300 puede realizar un procesamiento adicional, tal como realizar un proceso de desbloqueo para reducir los artefactos visuales a lo largo de los límites del bloque.

Esta divulgación generalmente puede referirse a “señalizar” cierta información, como elementos de sintaxis. El término “señalización” puede referirse generalmente a la comunicación de valores para elementos de sintaxis y/u otros datos usados para decodificar datos de video codificados. Es decir, el codificador de video 200 puede señalar valores para elementos de sintaxis en el flujo de bits. En general, la

señalización se refiere a generar un valor en el flujo de bits. Como se ha indicado anteriormente, el dispositivo fuente 102 puede transportar el flujo de bits al dispositivo de destino 116 sustancialmente en tiempo real, o no en tiempo real, tal como podría ocurrir cuando se almacenan elementos de sintaxis en el dispositivo de almacenamiento 112 para su posterior recuperación por el dispositivo de destino 116.

De acuerdo con las técnicas de esta divulgación, un método de codificación de datos de video incluye recibir una imagen de datos de video, reconstruir la imagen de datos de video y realizar un proceso de filtro basado en red neuronal (NN) sobre la imagen reconstruida de datos de video, en donde el proceso de filtro basado en NN comprende realizar una descomposición poliádica canónica (CP) con convolución separable.

Los estándares de codificación de video incluyen ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 o ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual e ITU-T H.264 (también conocido como ISO/IEC MPEG-4 AVC), codificación de video de alta eficiencia (HEVC, por sus siglas en inglés) o ITU-T H.265, que incluía su rango de extensión, extensión multivista (MV-HEVC, por sus siglas en inglés) y extensión escalable (SHVC, por sus siglas en inglés). La codificación de video versátil (VVC) o ITU-T H.266 ha sido desarrollada recientemente por el Joint Video

Expert TEAM (JVET) del Video Coding Experts Group (VCEG) de la ITU-T y Motion Picture Experts Group (MPEG) de ISO/IEC.

La especificación de VVC ha sido finalizada y denominada VVC FDIS en adelante, está disponible en
5 phenix.int-evry.fr/jvet/doc_end_user/documents/19_Teleconference/wg1
1/JVET-S2001-v17.

La Figura 2 es un diagrama conceptual que ilustra un marco de codificación de video híbrido. Los estándares de
10 codificación de video desde H.261 se han basado en el llamado principio de codificación de video híbrido, que se ilustra en la Figura 2. El término híbrido se refiere a la combinación de dos medios para reducir la redundancia en la señal de video, es decir, codificación de predicción y transformación
15 con cuantificación del residual de predicción. Mientras que la predicción y las transformaciones reducen la redundancia en la señal de video por decorrelación, la cuantificación disminuye los datos de la representación del coeficiente de transformación al reducir su precisión, idealmente al
20 eliminar solo detalles irrelevantes. Este principio de diseño de codificación de video híbrido también se usa en los dos estándares recientes, ITU-T H.265/HEVC e ITU-T H.266/VVC.

Como se muestra en la Figura 2, un codificador de video 130 híbrido moderno generalmente realiza particiones
25 de bloques, predicción interimagen o compensada por movimiento, predicción intraimagen, transformación,

cuantificación, codificación de entropía y filtrado post/en
bucle. En el ejemplo de la Figura 2, el codificador de video
130 incluye la unidad de suma 134, la unidad de transformación
136, la unidad de cuantificación 138, la unidad de
5 codificación por entropía 140, la unidad de cuantificación
inversa 142, la unidad de transformación inversa 144, la
unidad de suma 146, la unidad de filtro de bucle 148, el
búfer de imágenes decodificadas (DPB, por sus siglas en
inglés) 150, la unidad de intrapredicción 152, la unidad de
10 interpredicción 154 y la unidad de estimación de
movimiento 156.

En general, el codificador de video 130 puede,
cuando codifica datos de video, recibir datos de video de
entrada 132. La partición de bloques se usa para dividir
15 una imagen recibida (imagen) de los datos de video en
bloques más pequeños para la operación de los procesos de
predicción y transformación. Los primeros estándares de
codificación de video usaban un tamaño de bloque fijo,
típicamente 16×16 muestras. Los estándares recientes, tales
20 como HEVC y VVC, emplean estructuras de partición basadas
en árboles para proporcionar particiones flexibles.

La unidad de estimación de movimiento 156 y la
unidad de interpredicción 154 pueden predecir datos de video
de entrada 132, p. ej., a partir de datos previamente
25 decodificados de DPB 150. La predicción compensada por
movimiento o entre imágenes aprovecha la redundancia que

existe entre (de ahí “inter”) imágenes de una secuencia de video. De acuerdo con la compensación de movimiento basada en bloques, que se usa en los códecs de video modernos, la predicción se obtiene a partir de una o más imágenes
5 previamente decodificadas, es decir, la(s) imagen(es) de referencia. Las áreas correspondientes para generar la interpredicción se indican mediante información de movimiento, incluyendo vectores de movimiento e índices de imágenes de referencia.

10 En un códec de video reciente, se aplican estructuras de predicción jerárquica dentro de un grupo de imágenes (GOP) para mejorar la eficiencia de codificación. La Figura 3 ilustra un ejemplo de estructura de predicción jerárquica 162 con un tamaño de grupo de imágenes (GOP)
15 igual a 16.

Haciendo referencia nuevamente a la Figura 2, la predicción intraimagen explota la redundancia espacial que existe dentro de una imagen (de ahí “intra”) derivando la predicción para un bloque a partir de muestras espacialmente
20 vecinas (referencia) ya codificadas/decodificadas. La predicción angular direccional, la predicción DC y la predicción plana o plana se usan en el códec de video más reciente, incluyendo AVC, HEVC y VVC.

Los estándares de codificación de video híbridos
25 pueden aplicar una transformación en bloque al residuo de predicción (independientemente de si proviene de la

predicción inter o intraimagen). En los primeros estándares, incluyendo H.261/262/263, se emplea una transformación de coseno discreta (DCT, por sus siglas en inglés). En HEVC y VVC, se aplican más núcleos de transformación además de DCT para dar cuenta de diferentes estadísticas en la señal de video específica.

La cuantificación tiene como objetivo reducir la precisión de un valor de entrada o un conjunto de valores de entrada con el fin de disminuir la cantidad de datos necesarios para representar los valores. En la codificación de video híbrida, la cuantificación se aplica típicamente a muestras residuales transformaciones individuales, p. ej., para transformar coeficientes, dando como resultado niveles de coeficientes enteros. En los estándares de codificación de video recientes, el tamaño de paso se deriva de un llamado parámetro de cuantificación (QP, por sus siglas en inglés) que controla la fidelidad y la tasa de bits. Un tamaño de paso más grande reduce la velocidad de bits pero también deteriora la calidad, lo que, p. ej., da como resultado imágenes de video que exhiben artefactos de bloqueo y detalles borrosos.

La unidad de suma 134 puede calcular datos residuales como diferencias entre los datos de video de entrada 132 y los datos predichos de la unidad de intrapredicción 152 o la unidad de interpredicción 154. La unidad de suma 134 proporciona bloques residuales para la

unidad de transformación 136, que aplica una o más transformaciones al bloque residual para generar bloques de transformación. La unidad de cuantificación 138 cuantifica los bloques de transformación para formar
5 coeficientes de transformación cuantificados. Unidad de codificación por entropía 140 La entropía codifica los coeficientes de transformación cuantificados, así como otros elementos de sintaxis, tales como información de movimiento o información de intrapredicción, para generar
10 el flujo de bits de salida 158.

En tanto, la unidad de cuantificación inversa 142 cuantifica de manera inversa los coeficientes de transformación cuantificados, y la unidad de transformación inversa 144 transforma de manera inversa los coeficientes
15 de transformación, para reproducir bloques residuales. La unidad de suma 146 combina los bloques residuales con bloques de predicción (muestra por muestra) para producir bloques decodificados de datos de video. La unidad de filtro de bucle 148 aplica uno o más filtros (p. ej., al menos uno
20 de un filtro basado en red neuronal, un filtro de bucle basado en red neuronal, un filtro de postbucle basado en red neuronal, un filtro en bucle adaptativo o un filtro en bucle adaptativo predefinido) al bloque decodificado para producir bloques decodificados filtrados.

25 La unidad de codificación por entropía 140 puede realizar codificación aritmética binaria adaptativa al

contexto (CABAC, por sus siglas en inglés) en video codificado. CABAC se usa en códecs de video recientes, p. ej. AVC, HEVC y VVC, debido a su alta eficiencia.

La unidad de filtro de bucle 148 puede realizar un
5 filtrado posterior al bucle o en bucle. El filtrado post/en bucle es un proceso de filtrado (o combinación de tales procesos) que se aplica a la imagen reconstruida para reducir los artefactos de codificación. La entrada del proceso de filtrado es generalmente la imagen reconstruida, que es la
10 combinación de la señal residual reconstruida (que incluye error de cuantificación) y la predicción. Como se muestra en la Figura 2, las imágenes reconstruidas después del filtrado en bucle se almacenan en el búfer de imágenes decodificadas (DPB) 150 y se usan como referencia para la predicción
15 interimagen de imágenes posteriores. Los artefactos de codificación están determinados principalmente por el QP. Por lo tanto, la información de QP se usa generalmente en el diseño del proceso de filtrado. En HEVC, los filtros en bucle incluyen filtrado de desbloqueo y filtrado de desplazamiento
20 adaptativo de muestras (SAO). En el estándar VVC se introdujo un filtro de bucle adaptativo (ALF) como tercer filtro. El proceso de filtrado de ALF es como se muestra a continuación:

$$R'(i, j) = R(i, j) + \left(\left(\sum_{k \neq 0} \sum_{l \neq 0} f(k, l) \times K(R(i + k, j + l) - R(i, j), c(k, l)) + \right. \right. \\
25 \qquad \qquad \qquad \left. \left. 64 \right) \gg 7 \right), \quad (1)$$

donde can $R(i,j)$ es las muestras antes del proceso de filtrado, $R'(i,j)$ es el valor de la muestra después del proceso de filtrado. $f(k,l)$ denota los coeficientes de filtro, $K(x,y)$ es la función de recorte y $c(k,l)$ denota los parámetros de recorte. La variable k y l varía entre $-\frac{L}{2}$ y $\frac{L}{2}$ donde L denota la longitud del filtro. La función de recorte $K(x,y) = \min(y, \max(-y, x))$ que corresponde a la función *Clip3* $(-y, y, x)$. La operación de recorte introduce la no linealidad para hacer ALF más eficiente al reducir el impacto de los valores de muestra vecinos que son demasiado diferentes con el valor de muestra actual. En VVC, los parámetros de filtrado pueden señalizarse en el flujo de bits y seleccionarse de conjuntos de filtros predefinidos. El proceso de filtrado ALF también se puede resumir con la siguiente ecuación:

15

$$R'(i,j) = R(i,j) + ALF_residual_ouput(R), \quad (2)$$

La Figura 4 es un diagrama conceptual que ilustra un filtro basado en red neuronal 170 con cuatro capas. Diversos estudios han demostrado que la incrustación de redes neuronales en, p. ej., el marco de codificación de video híbrido de la Figura 2, puede mejorar la eficiencia de compresión. Las redes neuronales se han usado para la intrapredicción y la interpredicción para mejorar la eficiencia de predicción. El filtrado en bucle basado en

NN también es un tema de investigación prominente en los últimos años. En algunos ejemplos, el proceso de filtrado se aplica como un post-filtro. En tales ejemplos, el proceso de filtrado se aplica a la imagen de salida, y la imagen no filtrada puede usarse como imagen de referencia.

El filtro basado en NN 170 se puede aplicar además de los filtros existentes, tales como filtros de desbloqueo, desplazamiento adaptativo de muestras (SAO) y/o filtrado de bucle adaptativo (ALF). Los filtros basados en NN también se pueden aplicar exclusivamente, donde los filtros basados en NN están diseñados para reemplazar todos los filtros existentes. Adicionalmente, o alternativamente, los filtros basados en NN, tales como el filtro basado en NN 170, pueden diseñarse para complementar, mejorar o reemplazar cualquiera o todos los otros filtros.

El proceso de filtrado basado en NN de la Figura 4 puede tomar las muestras reconstruidas (p. ej., muestras de luma y croma que, en algunos ejemplos, pueden empaquetarse en un volumen 3D con 6 planos) como entradas, y las salidas intermedias son muestras residuales, que se añaden de nuevo a la entrada para refinar las muestras de entrada. El filtro basado en NN puede usar todos los componentes de color (p. ej., Y, U y V, o Y, Cb y Cr, p. ej., datos de luminancia 172A, crominancia de tono azul 172B y crominancia de tono rojo 172C) como entradas 172 para explotar las correlaciones entre componentes. Los diferentes componentes de color pueden

compartir los mismos filtros (incluyendo la estructura de red y los parámetros del modelo) o cada componente puede tener su(s) propio(s) filtro(s) específico(s).

El proceso de filtrado también se puede
5 generalizar de la siguiente manera:

$$R'(i,j) = R(i,j) + NN_filter_residual_ouput(R) \quad (3)$$

La estructura del modelo y los parámetros del
10 modelo del (de los) filtro(s) basado(s) en NN pueden predefinirse y almacenarse en el codificador de video 200 y en el decodificador de video 300. Los filtros también se pueden señalar en el flujo de bits.

En el ejemplo de la Figura 4, el filtro basado en
15 NN puede incluir una serie de capas de extracción de características, seguidas de una convolución de salida. En la Figura 4, las capas de extracción de características pueden incluir una capa de convolución (conv) 3x3 seguida de una capa de unidad lineal rectificada paramétrica (PReLU).
20 La capa de convolución aplica una operación de convolución a los datos de entrada, que implica un filtro o núcleo que se desliza sobre los datos de entrada (p. ej., las muestras de reconstrucción de la entrada 172) y el cálculo de productos de puntos en cada posición. La operación de convolución
25 captura esencialmente patrones locales dentro de los datos de entrada. Por ejemplo, en el contexto del procesamiento de

imágenes, estos patrones podrían ser bordes, texturas u otras características visuales. El filtro o núcleo es una pequeña matriz de pesos que se actualiza durante el proceso de entrenamiento. Al deslizar este filtro a través de los datos de entrada (o mapa de características de una capa anterior) y calcular el producto de puntos en cada posición, la capa de convolución crea un mapa de características que codifica jerarquías espaciales y patrones detectados en la entrada.

La salida de una capa de convolución es un conjunto de mapas de características, cada uno correspondiente a un filtro, capturando diferentes aspectos de los datos de entrada. Esta capa ayuda a la red neuronal a aprender características cada vez más complejas y abstractas a medida que los datos pasan a través de capas más profundas de la red. El primer 3x3 en la nomenclatura 3x3 conv 3x3x6x8 en la Figura 4 indica que la capa de convolución tiene un tamaño de filtro 3x3 (p. ej., una matriz 3x3). 3x3x6x8 se refiere tanto a las dimensiones de entrada como de salida de la capa de convolución, donde 6 es el número de canales de entrada, y 8 es el número de canales de salida.

La capa PReLU es una función de activación usada en redes neuronales, y se introdujo como una variante de la función de activación ReLU (Unidad Lineal Rectificada). Como se describió anteriormente, la capa de convolución genera mapas de características, cada uno correspondiente a un filtro, que representan las características detectadas

en la entrada. Siguiendo la capa de convolución, la capa PReLU aplica la función de activación de PReLU a cada elemento de los mapas de características producidos por la capa de convolución. Para valores positivos, la capa PReLU actúa como un ReLU estándar, pasando el valor a través de él. Para valores negativos, en lugar de establecerlos en cero (p. ej., como lo hace ReLU), la capa PReLU permite una salida pequeña, lineal y negativa. Esto mantiene activas las neuronas y mantiene el flujo de gradiente, lo que puede ser beneficioso para el aprendizaje en redes profundas.

En resumen, cuando una capa de convolución es seguida por una capa PReLU, la capa de convolución primero extrae características de los datos de entrada a través de un conjunto de filtros aprendidos. Los mapas de características resultantes se pasan después a través de la función de activación PReLU, que introduce la no linealidad y ayuda a evitar el problema de las neuronas moribundas al permitir un pequeño gradiente cuando las entradas son negativas. Esta combinación es efectiva para aprender patrones complejos en los datos mientras mantiene un flujo de gradiente robusto, especialmente beneficioso en arquitecturas de red más profundas.

Cuando se aplica filtrado basado en NN en la codificación de video, toda la señal de video (datos de píxeles) puede dividirse en múltiples unidades de procesamiento (p. ej., bloques 2D), y cada unidad de

procesamiento puede procesarse por separado o combinarse con otra información asociada con este bloque de píxeles. Las opciones posibles de una unidad de procesamiento incluyen un marco, una rebanada/baldosa, una CTU o cualquier forma y tamaño predefinido o señalizado. Por lo general, el filtrado basado en NN se realiza en bloques reconstruidos de datos de video. Aquí, los bloques y muestras reconstruidos pueden referirse tanto a bloques decodificados producidos por el decodificador de video 300, como a bloques reconstruidos en un bucle de reconstrucción del codificador de video 200.

Para mejorar aún más el rendimiento del filtrado basado en NN, diferentes tipos de datos de entrada pueden procesarse conjuntamente para producir la salida filtrada. Los datos de entrada pueden incluir, pero no se limitan a, píxeles/muestras de reconstrucción, píxeles/muestras de predicción, píxeles/muestras después del (de los) filtro(s) de bucle, información de estructura de partición, parámetros de desbloqueo (p. ej., fuerza límite (BS, por sus siglas en inglés)), valores QP, tipos de corte o imagen, o un mapa de modos de codificación o aplicabilidad de filtros. Los datos de entrada se pueden proporcionar en diferentes granularidades. Las muestras de reconstrucción y predicción de luma podrían proporcionarse con la resolución original, mientras que las muestras de croma podrían proporcionarse a una resolución más baja, p. ej., para una representación 4:2:0, o pueden muestrearse hacia arriba a la resolución de

Luma para lograr una representación por píxel. De manera similar, la información de modo QP, BS, partición o codificación puede proporcionarse a una resolución más baja, incluyendo casos con un único valor por trama, corte o bloque de procesamiento (p. ej., QP). En otros ejemplos, la información de QP, BS, partición o modo de codificación puede expandirse (p. ej., replicarse) para lograr una representación por píxel/muestra.

Un ejemplo de una arquitectura que utiliza datos suplementarios se muestra en la Figura 5. La Figura 5 es un diagrama en bloque que ilustra un filtro basado en CNN de ejemplo con muestras de entrada acojinadas y datos suplementarios. El filtro 171 basado en NN usa píxeles/muestras del bloque de procesamiento combinados con datos suplementarios como entrada 174. La entrada 174 puede incluir 4 subbloques de muestras luma entrelazadas (Yx4) 174A y datos de crominancia de tono azul (U) 174B asociados y datos 174C de crominancia de tono rojo (V). Los datos suplementarios incluyen una etapa 176 del parámetro de cuantificación (QP) y una fuerza límite (BS) 178. El área de los píxeles/muestras de entrada se puede ampliar con 4 píxeles/muestras acojinadas de cada lado. Las dimensiones resultantes del volumen de procesamiento son $(4+64+4) \times (4+64+4) \times (4Y+2UV+1QP + 3BS)$.

En relación con el filtro basado en NN en la Figura 4, el filtro basado en NN 171 puede incluir dos o más capas

ocultas que utilizan tanto convoluciones 1x1 como una capa ReLU con fugas. Una capa de ReLU con fugas. Al igual que una capa PReLU, una capa ReLU con fugas permite que se emita un gradiente pequeño, distinto de cero, cuando la capa no está activa. En lugar de dar salida a cero para entradas negativas, el ReLU con fugas multiplica estas entradas por una pequeña constante. Esta pequeña pendiente asegura que incluso las neuronas que de cualquier otra manera estarían inactivas aun contribuyan una pequeña cantidad al aprendizaje de la red, reduciendo la probabilidad de que el problema de ReLU muera.

Para mejorar aún más el rendimiento del filtrado basado en NN, se pueden diseñar soluciones multimodo. Por ejemplo, para cada unidad de procesamiento, el codificador de video 200 puede seleccionar entre un conjunto de modos basados en la optimización de la distorsión de la tasa y la elección puede señalizarse en el flujo de bits. Los diferentes modos pueden incluir diferentes modelos NN, diferentes valores que se usan como información de entrada de los modelos NN, etc. En un ejemplo, el codificador de video 200 y el decodificador de video 300 pueden usar una solución de filtrado basado en NN con múltiples modos basados en un único modelo NN usando diferentes valores QP como entrada del modelo NN para diferentes modos.

A continuación se describen ejemplos de arquitecturas de ILF de CNN que están siendo desarrolladas por JVET. Una de esas arquitecturas es un ILF de CNN

multimodo con un bloque de atención. En JVET-Z0113, Y. Li, K. Zhang, L. Zhang, H. Wang, M. Coban, A. M. Kotra, M. Karczewicz, F. Galpin, K. Andersson, J. Ström, D. Liu, R. Sjöberg, EE1-1.7: Combined Test of EE1-1.6 and EE1-1.3, 5 JVET-Z0113, April 2022 (de aquí en adelante “JVET-Z0113”), se describe una solución de filtrado basado en NN con múltiples modos, como se describió anteriormente, con respecto al filtrado basado en NN con un diseño multimodo.

En un ejemplo, se puede usar una solución de 10 filtrado basado en NN con múltiples modos, como se describió anteriormente. La estructura de la red neuronal se muestra en la Figura 6. En la primera parte de la Figura 6, los diferentes tipos de datos de entrada se convolucionan con un número de núcleos de tamaño 3x3 (conv3x3) para producir 15 mapas de características, las entradas convolucionadas sufren activación por unidades lineales rectificadas paramétricas (PReLU), y después los resultados para cada tipo de datos son concatenados (concat), fusionados y submuestreados una vez (transición) para crear la salida y. 20 Esta salida se alimenta entonces a través de bloques residuales de atención (AttRes), cada uno con la estructura mostrada en la Figura 6. La salida del último bloque residual de atención z se alimenta a la última parte de la red.

Por ejemplo, el filtro basado en NN de la Figura 25 6 incluye una primera porción que incluye convoluciones de entrada 3x3 510A-510E y respectivas unidades lineales

rectificadas paramétricas (PReLU) 512A-512E para cada una de las entradas para generar mapas de características (p. ej., la sección de extracción de características del filtro NN). La unidad de concatenación 514 concatena los mapas de
5 características y los proporciona al bloque de fusión 516 y al bloque de transición 522. El filtro basado en NN en la Figura 6 incluye además un conjunto 528 de bloques residuales de atención (AttRes) 530A-530N; y una última porción (p. ej., la sección de cola) que incluye
10 convolución de 3x3 550, PReLU 552, convolución 3x3 554 y unidad de mezcla de píxeles 556. Los bloques AttRes también pueden denominarse bloques troncales.

En la primera porción (p. ej., la sección de extracción de características), se reciben diferentes
15 entradas, incluyendo el parámetro de cuantificación (QP) 500, información de partición (parte) 502, fuerza límite (BS) 504, muestras de predicción (pred) 506 y muestras de reconstrucción (rec) 508. Las respectivas convoluciones 3x3 510A-510E y PReLU 512A-512E convolucionan y activan
20 las entradas respectivas para producir mapas de características. A continuación, la unidad de concatenación 514 concatena los mapas de características. El bloque de fusión 516, que incluye la convolución 1x1 518 y PReLU 520, fusiona los mapas de características
25 concatenadas. El bloque de transición, que incluye la convolución 3x3 524 y PReLU 526, submuestrea las entradas

fusionadas para crear la salida 188. La salida 188 se alimenta entonces a través del conjunto 528 de los bloques residuales de atención 530A-530N, que pueden incluir un número diverso de bloques residuales de atención, p. ej.,

5 8. El bloque de atención se explica adicionalmente con respecto a la Figura 7. La salida 189 del último del conjunto 528 de los bloques residuales de atención 530 se alimenta a la última porción del filtro basado en NN. En la última porción, que puede ser un bloque de cola, la

10 convolución 3x3 550, PReLU 552, convolución 3x3 554 y la unidad de mezcla de píxeles 556 procesa la salida 189, y la unidad de adición 558 combina este resultado con las muestras de reconstrucciones de entrada originales 508. En última instancia, esto forma la salida filtrada para

15 presentación y almacenamiento como referencia para la interpredicción posterior, p. ej., en un búfer de imágenes decodificadas (DPB). En algunos ejemplos, el filtro basado en NN de la Figura 6 usa 96 mapas de características.

La Figura 7 es un diagrama conceptual que ilustra

20 un bloque residual de atención de la Figura 6. Es decir, la Figura 7 representa el bloque residual de atención 530, que puede incluir componentes similares a los de los bloques residuales de atención 530A-530N de la Figura 6. En este ejemplo, el bloque residual de atención 530 incluye

25 la primera convolución 3x3 532, el filtro de unidad lineal rectificadora paramétrica (PReLU) 534, la segunda

convolución 3x3 536, un bloque de atención 538 y la unidad de adición 540. La unidad de adición 540 combina la salida del bloque de atención 538 y la salida 188, inicialmente recibida por convolución 532, para generar la salida 189.

5 La Figura 8 es un diagrama conceptual que ilustra una capa de atención espacial de ejemplo de la Figura 7. Como se muestra en la Figura 8, una capa de atención espacial del bloque residual de atención 530 incluye convolución 3x3 706, PReLU 708, convolución 3x3 710, unidad de expansión de tamaño 712, convolución 3x3 720, PReLU 722 y convolución 3x3 724. La convolución 3x3 706 recibe entradas 702, correspondientes al parámetro de cuantificación (QP) 500, información de partición (parte) 502, fuerza límite (BS) 504, información de predicción (pred) 506, y muestras
10 reconstruidas (rec) 508 de la Figura 6. La convolución 3x3 720 recibe Z_K 704. Las salidas de la unidad de expansión de tamaño 712 y convolución 3x3 724 se combinan, y después se combinan con el valor R 730 para generar el valor S 732. El valor S 732 se combina entonces con el valor Z_K 704 para
15 generar el valor Z_{K+1} de salida 734.

Otro ejemplo de arquitectura NN es un ILF de CNN multimodo de complejidad reducida. En JVET-AC0155, S. Eadie, M. Coban, M. Karczewicz, EE1-1.9: Reduced complexity CNN-based in-loop filtering, JVET-AC0155, enero
25 de 2023, se propuso un diseño alternativo de la arquitectura NN. Por ejemplo, se puede usar un mayor número

de bloques residuales de baja complejidad en el troncal del filtro del ejemplo de la Figura 6, junto con un número reducido de canales (mapas de características) y la eliminación de los módulos de atención. La estructura de filtrado basado en CNN propuesta (p. ej., para el filtrado de luma) se muestra en la Figura 9.

La Figura 9 es un diagrama en bloque que ilustra un ejemplo de una arquitectura de filtro basada en CNN simplificada con muestras de entrada acojinadas y datos suplementarios. El filtro basado en NN de la Figura 9 incluye las convoluciones 3x3 810A-810E y PReLU 812A-812E, que convolucionan las entradas correspondientes, es decir, QP 800, Parte 802, BS 804, Pred 806 y Rec 808 para generar mapas de características (p. ej., la sección de extracción de características). La unidad de concatenación 814 concatena las entradas convolucionadas (p. ej., los mapas de características). A continuación, el bloque de fusión 816 fusiona los mapas de características concatenadas usando convolución 1x1 818 y PReLU 820. A continuación, el bloque de transición 822 procesa los datos fusionados usando la convolución 3x3 824 y PReLU 826.

En este ejemplo, el filtro basado en NN incluye un conjunto 828 de bloques residuales 830A-830N (también llamados bloques troncales), cada uno de los cuales puede estar estructurado de acuerdo con la estructura de bloque residual 830 de la Figura 10, como se analiza a

continuación. Los bloques residuales 830A-830N pueden reemplazar los bloques AttRe 530A-530N de la Figura 6. El ejemplo de la Figura 9 puede usarse para el filtrado de luminancia (luma), aunque como se analiza a continuación, se pueden hacer modificaciones similares para el filtrado de crominancia (croma).

El número de bloques y canales residuales incluidos en el conjunto 828 de la Figura 9 se puede configurar de manera diferente. Es decir, N se puede establecer en un valor diferente, y el número de canales en la estructura de bloques residual 830 se puede establecer en un número diferente de 160, para lograr diferentes compensaciones de rendimiento y complejidad. El filtrado de croma se puede realizar con estas modificaciones para el procesamiento de canales de croma.

El conjunto 828 de bloques residuales 830A-830N tiene N instancias de estructura de bloque residual 830. En un ejemplo, N puede ser igual a 32, de tal manera que hay 32 estructuras de bloques residuales. Los bloques residuales 830A-830N pueden usar 64 mapas de características, lo que se reduce en relación con los 96 mapas de características usados en el ejemplo de la Figura 6.

En la última porción de la Figura 9, la convolución 3x3 850, PReLU 852, la convolución 3x3 854 y la unidad de mezcla de píxeles 856 procesa la salida del conjunto 828, y la unidad de adición 858 combina este resultado con las muestras de reconstrucciones de entrada

originales (REC) 808. En última instancia, esto forma la salida filtrada para presentación y almacenamiento como referencia para la interpredicción posterior, p. ej., en un búfer de imágenes decodificadas (DPB).

5 En un ejemplo, el número de bloques residuales y mapas de características (convoluciones) usados es $M=24$ y 64, respectivamente, en el ejemplo de la Figura 9. En los bloques residuales 830A-830N, el número de canales primero sube a 160 antes de la capa de activación y después baja
10 a 64 después de la capa de activación. El número de bloques y canales residuales se puede configurar de manera diferente (M se establece en otro valor y el número de canales en el bloque residual se puede establecer en un número diferente de 160) para diferentes compensaciones de
15 rendimiento y complejidad. El filtrado de croma puede seguir el concepto de la Figura 6, con las modificaciones descritas anteriormente en su troncal para el procesamiento de canales de croma.

 En aun otra arquitectura NN, los bloques residuales
20 pueden ser reemplazados por bloques de filtro como se muestra en la Figura 11, que se analiza más adelante en esta divulgación. En este ejemplo, se elimina la rama de derivación alrededor de las capas de convolución y activación en el bloque residual de la solución anterior. El número de
25 canales y el número de bloques de filtro pueden ser configurables, por ejemplo, 64 canales, 24 bloques de filtro,

con 160 canales antes y después de la activación, lo que da como resultado que la complejidad de la red sea 605,93 kMAC y el número de parámetros es 1,5 M para el modelo intra luma.

Otro ejemplo de arquitectura NN es un ILF de CNN multimodo con convolución separable. Se puede lograr una mayor reducción de la complejidad de la arquitectura de ILF de ILF con la utilización de la convolución separable en lugar de convoluciones 2D (3x3). En JVET-AD0023, prueba EE1 1.3.5, una aproximación de convolución de rango bajo descompone una convolución $3 \times 3 \times M \times N$ en una convolución pixelada ($1 \times 1 \times M \times R$), se aplicaron dos convoluciones separables ($3 \times 1 \times R \times R$, $1 \times 3 \times R \times R$) y otra convolución por píxeles ($1 \times 1 \times R \times N$) a un bloque residual de la arquitectura descrita en JVET-AC0155. En este ejemplo, R es el rango de la aproximación, y puede ablacionar el rendimiento/complejidad de la aproximación.

La Figura 10 es un diagrama conceptual que ilustra un ejemplo de estructura de bloque residual 830 de la Figura 9. En este ejemplo, la estructura de bloque residual 830 incluye la primera convolución 1×1 832, que puede aumentar un número de canales de entrada a 160, antes de que una capa de activación (PReLU 834) procese los canales de entrada. PReLU 834 puede reducir de esta manera el número de canales a 64 a través de este procesamiento. La segunda convolución 1×1 836 procesa entonces los canales reducidos, seguida de la convolución 3×3 838. Finalmente,

la unidad de combinación 840 puede combinar la salida de convolución 3x3 838 con la entrada original recibida por la estructura de bloques residual 830.

En aun otra arquitectura NN, los bloques
5 residuales pueden ser reemplazados por bloques de filtro (también llamados bloques troncales) como se muestra en la Figura 11. En este ejemplo, se elimina la rama de derivación alrededor de todas las capas en cada bloque residual, como se muestra en la Figura 12. El número de
10 canales y número de bloques de filtro es configurable, por ejemplo, 64 canales, 24 bloques de filtro, con 160 canales antes y después de la activación. En algunos ejemplos, la complejidad de una red de este tipo es 605,93 kMAC y el número de parámetros es 1,5 M para un modelo intra luma.

15 La Figura 11 es un diagrama conceptual que ilustra otro ejemplo de estructura de bloques de filtrado que puede sustituirse por el conjunto de bloques residuales de atención de la Figura 6 de acuerdo con las técnicas de esta divulgación. El filtro basado en NN de la Figura 11
20 incluye convoluciones 3x3 1010A-1010E y PReLU 1012A-1012E, que convolucionan entradas respectivas, es decir, QP 1000, Parte 1002, BS 1004, Pred 1006 y Rec 1008 para formar mapas de características (p. ej., la sección de extracción de características). La unidad de concatenación 1014
25 concatena los mapas de características. A continuación, el bloque de fusión 1016 fusiona las entradas concatenadas

usando convolución 1x1 1018 y PReLU 1020. A continuación, el bloque de transición 1022 procesa los datos fusionados usando la convolución 3x3 1024 y PReLU 1026.

En este ejemplo, la unidad de filtrado basado en
5 NN incluye un conjunto 1028 de N bloques de filtro 1030A-1030N (también denominados bloques troncales), cada uno de los cuales puede tener la estructura del bloque de filtro 1030 de la Figura 12 como se analiza a continuación. La estructura de bloque de filtro 1030 puede ser
10 sustancialmente similar a la estructura de bloque residual 830, excepto que la unidad de combinación 840 se omite de la estructura de bloque de filtro 1030, de tal manera que la entrada no se combina con la salida. En su lugar, la salida de cada estructura de bloque residual puede ser
15 alimentada directamente al bloque posterior.

En la última porción de la Figura 11, la convolución 3x3 1050, PReLU 1052, convolución 3x3 1054 y la unidad de mezcla de píxeles 1056 procesa la salida del conjunto 1028, y la unidad de adición 1058 combina este
20 resultado con las muestras de reconstrucciones de entrada originales (REC) 1008. En última instancia, esto forma la salida filtrada para presentación y almacenamiento como referencia para la interpredicción posterior, p. ej., en un búfer de imágenes decodificadas (DPB).

25 En un ejemplo, la arquitectura de la Figura 11, se implementa con los parámetros $K=64$, $M=160$ y $R=51$, y el

número total de 24 bloques residuales da como resultado que la complejidad de la red sea 356,43 kMAC y el número de parámetros es 1,07 M para el modelo intra luma.

La Figura 12 es un diagrama conceptual que ilustra un ejemplo de estructura de bloque de filtro 1030 de la Figura 11. En este ejemplo, la estructura de bloque de filtro 1030 incluye la primera convolución 1x1 1032, que puede aumentar un número de canales de entrada a 160, antes de que una capa de activación (PReLU 1034) procese los canales de entrada. De esta manera, PReLU 1034 puede reducir el número de canales a 64 a través de este procesamiento. La segunda convolución 1x1 1036 procesa entonces los canales reducidos, seguida de la convolución 3x3 1038. Como se analizó anteriormente, la estructura de bloque de filtro 1030 no incluye una unidad de combinación, en contraste con la estructura de bloque residual 830 de la Figura 10.

La convolución con un núcleo 3x3 es popular en los filtros basados en NN. En las arquitecturas descritas anteriormente, se utiliza una convolución 3X3xNxM en múltiples secciones y bloques, con un núcleo 3x3 deslizándose en el dominio espacial (2D). Sin embargo, la convolución multidimensional, tal como una convolución de núcleo 2D, introduce una complejidad significativa. De acuerdo con las técnicas de esta divulgación, el codificador de video 200 y el decodificador de video 300 pueden configurarse para utilizar convoluciones separables en lugar de, o en lugar

de, una convolución multidimensional (p. ej., una convolución $3 \times 3 \times N \times M$). Por ejemplo, se pueden usar dos convoluciones unidimensionales separables en lugar de una convolución 3×3 en cualquier sección de un filtro basado en NN. El uso de
5 convoluciones separables puede reducir la complejidad de cálculo y los requerimientos de ancho de banda de memoria.

Para evitar un cálculo excesivo y reducir los conjuntos de parámetros que se originan a partir de convoluciones multidimensionales, tales como convoluciones
10 3×3 (o componentes de convolución 2D de núcleos de mayor dimensionalidad) en las arquitecturas CNN descritas anteriormente o similares, esta divulgación describe técnicas en las que el codificador de video 200 y el decodificador de video 300 están configurados para utilizar
15 convoluciones separables (p. ej., convolución separable 1D) producidas por aproximación de baja complejidad en lugar de multidimensional (p. ej., 2D) convoluciones que se deslizan en la dirección espacial. Si bien las técnicas de esta divulgación se describen con referencia a convoluciones 3×3 ,
20 las técnicas de descomposición de esta divulgación pueden usarse para cualquier tamaño de convoluciones multidimensionales (p. ej., 4×4 , 5×5 o más grandes). En general, una convolución multidimensional tiene un tamaño de núcleo de $n_1 \times n_2$ en dimensión espacial donde n_1 y n_2 son
25 enteros positivos. Los valores de n_1 y n_2 pueden ser iguales o diferentes. La convolución multidimensional puede tener

además un tamaño de K en una dimensión de profundidad (p. ej., $n_1 \times n_2 \times K$). Además, con el número de canales de salida M , la convolución multidimensional se puede expresar como un tensor 4-D de $n_1 \times n_2 \times K \times M$.

5 En un ejemplo de la divulgación, una aproximación de convolución de rango bajo descompone una convolución $3 \times 3 \times M \times N$ en una convolución pixelada ($1 \times 1 \times M \times R$), dos convoluciones separables ($3 \times 1 \times R \times R$, $1 \times 3 \times R \times R$) y otra convolución por píxeles ($1 \times 1 \times R \times N$). Aquí, R es el rango de
10 la aproximación, y se puede usar para ajustar el rendimiento/complejidad de la aproximación. El valor de R puede ser un número entero. En algunos ejemplos, R puede derivarse como una función (relación) de M o N , o $\max.(M, N)$. En algunos ejemplos, R se puede establecer igual a $A \times$
15 $\max.(M, N)$, siendo A menor que 1 (p. ej., 0,2, 0,5, 0,8), A siendo mayor que 1 (p. ej., 1,0, 1,2) u otros valores.

En un ejemplo general, una convolución multidimensional puede aproximarse mediante una pluralidad de convoluciones separables realizando una primera
20 convolución de tamaño $n_1 \times 1$ y realizando una segunda convolución de tamaño $1 \times n_2$ en la salida de la primera convolución.

La Figura 13 es un diagrama en bloque que ilustra una red troncal de extracción de características multiescala
25 de ejemplo con convolución de dos componentes. El ejemplo de la Figura 13 puede ser similar al ejemplo de la Figura 10

con el uso de una aproximación de una convolución $3 \times 3 \times K \times K$ con una convolución $3 \times 1 \times K \times R$ y una convolución $1 \times 3 \times R \times K$.

En el ejemplo de la Figura 13, el bloque residual 1420 incluye una convolución $1 \times 1 \times K \times M$ 1402, seguida de PReLU 1404. La salida de PReLU 1404 se introduce en la convolución $1 \times 1 \times M \times K$ 1406. Una convolución $3 \times 3 \times K \times K$ 1408 del bloque residual 1420 se aproxima por una convolución $3 \times 1 \times K \times R$ 1400 y después una convolución $1 \times 3 \times R \times K$ 1410. La salida de la convolución $1 \times 3 \times R \times K$ 1410 puede ingresarse a la unidad de combinación 1412 que puede combinar la salida de $1 \times 3 \times R \times K$ convolución 1410 con una entrada a la convolución $1 \times 1 \times K \times M$ 1402. R es el rango canónico de la descomposición. Un rango más bajo implica una mayor reducción de la complejidad.

La Figura 14 es un diagrama en bloque que ilustra un ejemplo de filtro unificado con un modelo unido (luma y croma unidos). La extracción de características multiescala con una red de convolución de dos componentes ilustrada en la Figura 14 se propone en Y. Li, S. Eadie, D. Rusanovskyy, M. Karczewicz, EE1-Related: Combination test of EE1-1.3.5 and multiscale component of EE1-1.6, JVET-AD0211, Abril de 2023 (JVET-AD0211). Las convoluciones 3×3 pueden descomponerse en una convolución $3 \times 1 \times C1 \times R$ y seguidas de una convolución $1 \times 3 \times R \times C2$, donde $C1$ y $C2$ son el número de canales de entrada y salida, respectivamente, y R es el rango de la aproximación (ver V. Lebedev, Y. Ganin, M. Rakhuba, I. Oseledets, V. Lempitsky, Speeding up Convolution Neural

Networks Using Fine-tuned CP-Decomposition, ICLR 2015, Disponible en línea: <https://arxiv.org/pdf/1412.6553>). En algunos ejemplos, el parámetro R puede hacerse proporcional a $R=C1 \times C2 / (C1+C2)$ y controla la complejidad de la
5 aproximación.

La Figura 14 muestra una arquitectura propuesta con bloques de convolución 3×3 reemplazados por convoluciones separables de 3×1 y 1×3 . En este ejemplo, la estructura de bloque residual 1430 incluye una primera
10 convolución 1×1 1432 antes de una primera capa de activación (PReLU 1434) y, en paralelo con la primera convolución 1×1 1432 y PReLU 1434, una convolución 3×3 1440 y una segunda capa de activación (PReLU 1442). Una segunda convolución 1×1 1436 procesa entonces la salida combinada de PReLU 1434 y
15 PReLU 1442, seguida de la convolución 3×3 1438. En el ejemplo de la Figura 14, sin embargo, la convolución 1440 3×3 puede aproximarse usando una pluralidad de convoluciones separables, mostradas como convolución 3×1 1450 y convolución 1×3 1452 en la Figura 14. De manera similar, la
20 convolución 3×3 1438 puede aproximarse usando una pluralidad de convoluciones separables, mostradas como convolución 3×1 1460 y convolución 1×3 1462 en la Figura 14.

A modo de ejemplo, la arquitectura ilustrada en la Figura 14 puede implementarse con los parámetros $R1=8$,
25 $R2=44$, $M1=160$ y $M2=16$, y el número total de 24 bloques

residuales, la complejidad de la red será de 358,43 kMAC y el número de parámetros es 1,07 M para el modelo intra luma.

La Figura 15 es un diagrama en bloque que ilustra un filtro unificado de ejemplo con un modelo unido (luma y cromas unidos). El filtro basado en NN de la Figura 15 puede incluir IPB 1500 (que proporciona información con respecto a si un bloque está inter o intrapredicho), QP_{SLICE} 1502 (que proporciona un parámetro de cuantificación para un corte), QP_{BASE} 1504 (que proporciona un parámetro de cuantificación para la secuencia), fuerza límite (BS) 1506, muestras de predicción (PRED) 1508 y muestras de reconstrucción (REC) 1509. El decodificador de video 300 puede aplicar una convolución 3x3 correspondiente de convoluciones 3x3 1510A-1510F a cada una de las entradas. El decodificador de video 300 puede aplicar una PReLU correspondiente de PReLUs 1512A-1512F a las salidas de convoluciones 3x3 1510A-1510F. En algunos ejemplos, los bloques troncales 1530A-1530N pueden ser bloques residuales.

El bloque de fusión 1516 puede fusionar los mapas de características usando convolución 1x1 1518 y PReLU 1520. A continuación, el bloque de transición 1522 procesa los datos fusionados usando la convolución 3x3 1524 y PReLU 1526.

El decodificador de video 300 puede aplicar un conjunto 1528 de bloques troncales. El conjunto 1528 puede incluir una pluralidad de bloques de filtro 1530A-1530N (también denominados en la presente descripción bloques

troncales). En algunos ejemplos, $N=24$, de tal manera que hay 24 bloques de filtro. El decodificador de video 300 puede aplicar la convolución 3×3 1550 a la salida del conjunto 1528. El decodificador de video 300 puede aplicar

5 PReLU 1552 a la salida de la convolución 3×3 1550. El decodificador de video 300 puede aplicar la convolución 3×3 1554 a la salida de PReLU 1552. El decodificador de video 300 puede recortar 1558 la salida de la convolución 3×3 1554 para generar muestras UV reconstruidas filtradas

10 de la imagen. El decodificador de video 300 puede realizar una mezcla de píxeles 1556 en la salida de la convolución 3×3 1554 y recortar 1560 la salida de la mezcla de píxeles 1556 para generar muestras Y reconstruidas filtradas de la imagen. Por ejemplo, la mezcla de píxeles 1556 puede

15 aumentar su muestra de entrada de tal manera que la salida de la mezcla de píxeles 1556 tenga un tamaño de $w \times 2$, $h \times 2$, $c/4$, donde w es el ancho, h es la altura de entrada y c es el número de entrada de canales de la entrada a la mezcla de píxeles 1556. En algunos ejemplos, los parámetros del

20 ejemplo de la Figura 15 incluyen uno o más de: $d_1 = 192$, $d_2 = 32$, $d_3 = 16$, $d_4 = 16$, $d_5 = 16$, $C = 64$, y $d_6 = 48$.

La Figura 16 es un diagrama en bloque que ilustra un bloque troncal de ejemplo. El bloque troncal 1600 de la Figura 16 puede ser un ejemplo de cualquiera de los bloques

25 de filtro 1530A-1530N. El bloque troncal 1600 puede incluir una rama multiescala 1620 donde las convoluciones se realizan

en paralelo. Por ejemplo, una convolución 1x1 1602 puede realizarse en paralelo con una convolución 3x1 1604 seguida de una convolución 1x3 1606. La rama multiescala 1620 puede realizar extracción de características multiescala.

5 El decodificador de video 300 puede aplicar una PReLU 1608 a la rama multiescala de salida 1620. El decodificador de video 300 puede aplicar una convolución 1x1 1610 a la salida de PReLU 1608. El decodificador de video 300 puede aplicar una convolución 1x3 1612 a la
10 salida de convolución 1x1 1610. El decodificador de video 300 puede aplicar una convolución 3x1 1614 a la salida de convolución 1x3 1612. En algunos ejemplos, los parámetros del ejemplo de la Figura 16 incluyen $C = 64$, $C_1 = 160$, $C_{21} = 32$, $C_{22} = 32$, y $C_{31} = 64$.

15 La Figura 17 es un diagrama en bloque que ilustra el filtro unificado de ejemplo con modelos separados de luma y croma (luma). El filtro basado en NN de la Figura 17 puede incluir IPB 1700, QP_{BASE} 1702, BS 1704, PRED 1706, y muestras de luma reconstruidas (REC_{EXT} Y) 1708. El
20 decodificador de video 300 puede aplicar una convolución 3x3 correspondiente de convoluciones 3x3 1710A-1710E a cada una de las entradas. El decodificador de video 300 puede aplicar una PReLU correspondiente de PReLU 1712A-1712E a las salidas de convoluciones 3x3 1710A-1710E.

25 El bloque de fusión 1716 puede fusionar los mapas de características usando convolución 1x1 1718 y PReLU 1720.

A continuación, el bloque de transición 1722 procesa los datos fusionados usando la convolución 3x3 1724 y PReLU 1726.

El decodificador de video 300 puede aplicar un conjunto 1728 de bloques troncales. El conjunto 1728 puede
5 incluir una pluralidad de bloques de filtro 1730A-1730N, que también pueden denominarse bloques troncales. En algunos ejemplos, $N=20$, de tal manera que hay 20 bloques de filtro. El decodificador de video 300 puede aplicar la convolución 3x3 1750 a la salida del conjunto 1728. El
10 decodificador de video 300 puede aplicar PReLU 1752 a la salida de la convolución 3x3 1750. El decodificador de video 300 puede aplicar la convolución 3x3 1754 a la salida de PReLU 1752. El decodificador de video 300 puede recortar 1758 la salida de convolución 3x3 1754 para generar
15 muestras UV reconstruidas filtradas de la imagen. El decodificador de video 300 puede realizar una mezcla de píxeles 1756 en la salida de la convolución 3x3 1754 y recortar 1760 la salida de la mezcla de píxeles 1756 para generar muestras Y reconstruidas filtradas de la imagen.

20 Uno o más de los bloques de filtro 1730A-1730N del conjunto 1728 pueden ser ejemplos del bloque troncal 1600 de la Figura 16. En algunos ejemplos, los parámetros del ejemplo de la Figura 17 junto con el ejemplo de la Figura 16 pueden incluir uno o más de: $d_1 = 192$, $d_2 = 32$,
25 $d_3 = 16$, $d_4 = 16$, $d_5 = 16$, $C = 64$, $C_1 = 160$, $C_{21} = 32$, $C_{22} = 32$, $C_{31} = 64$, $N = 20$, y $d_6 = 48$.

La Figura 18 es un diagrama en bloque que ilustra un filtro unificado de ejemplo con modelos separados de luma y croma (croma). Las entradas para el filtro basado en NN de la Figura 18 pueden incluir REC_{EXT} Y 1800, que
5 pueden ser muestreados por un factor de 2, QP_{BASE} 1804, BS 1806, PRED 1808, y muestras de croma reconstruidas (REC_{EXT} UV) 1809. El decodificador de video 300 puede aplicar una convolución 3x3 correspondiente de convoluciones 3x3 1810A-1810E a cada una de las entradas. El decodificador de video
10 300 puede aplicar una PReLU correspondiente de PReLUs 1812A-1812E a las salidas de convoluciones 3x3 1810A-1810E.

El bloque de fusión 1816 puede fusionar los mapas de características usando convolución 1x1 1818 y PReLU 1820. A continuación, el bloque de transición 1822 procesa los
15 datos fusionados usando la convolución 3x3 1824 y PReLU 1826.

El decodificador de video 300 puede aplicar un conjunto 1828 de bloques troncales. El conjunto 1828 puede incluir una pluralidad de bloques de filtro 1830A-1830N, que también pueden denominarse bloques troncales. En
20 algunos ejemplos, $N=16$, de tal manera que hay 24 bloques de filtro. El decodificador de video 300 puede aplicar la convolución 3x3 1850 a la salida del conjunto 1828. El decodificador de video 300 puede aplicar PReLU 1852 a la salida de la convolución 3x3 1850. El decodificador de video
25 300 puede aplicar la convolución 1854 3x3 a la salida de PReLU 1852. El decodificador de video 300 puede recortar

1858 la salida de la convolución 3x3 1854 para generar
muestras UV reconstruidas filtradas de la imagen. El
decodificador de video 300 puede realizar una mezcla de
píxeles 1856 en la salida de la convolución 3x3 1854 y
5 recortar 1860 la salida de la mezcla de píxeles 1856 para
generar muestras Y reconstruidas filtradas de la imagen.

Uno o más de los bloques de filtro 1830A-1830N
del conjunto 1828 pueden usar la arquitectura de bloques
troncales de la Figura 16. En algunos ejemplos, los
10 parámetros del ejemplo de la Figura 18 junto con la Figura
16 pueden incluir uno o más de: $d_1 = 192$, $d_2 = 32$, $d_3 = 16$,
 $d_4 = 16$, $d_5 = 16$, $C = 64$, $C_1 = 160$, $C_{21} = 32$, $C_{22} = 32$, C_{31}
 $= 64$, $N = 16$, y $d_6 = 48$.

Se ha integrado un troncal de extracción de
15 características multiescala con descomposición de dos
componentes en un modelo unificado en EE. Además, la
especificación del EE incluye dos versiones del modelo, que
son 1) un modelo unificado para luma y croma unidos, como
se muestra en la Figura 15, y 2) modelos separados para
20 luma y croma, respectivamente, como se muestra en las
Figuras 17 y 18.

La Arquitectura de Filtro Unificado (UF, por sus
siglas en inglés) adoptada por JVET, (como se muestra en las
Figuras 15-18), apunta a un nivel relativamente de alto
25 rendimiento de aplicaciones de codificación de video,
operando a más de 400 kMAC/píxel para la codificación de

datos YCbCr 4:2:0. Para ejecutar el filtro en bucle con un códec de video en dispositivos computacionalmente de última generación y/o con restricción de energía, la complejidad de las herramientas basadas en NN está limitada a niveles inferiores a 20 kMAC/píxel. Por lo tanto, la Arquitectura UF de las Figuras 15-18 no puede implementarse en tales dispositivos computacionalmente y/o con restricción de potencia.

Para lograr un nivel de complejidad requerido (p. ej., <20 kMAC/píxel) y preservar una parte relativamente grande del rendimiento de codificación que brinda la arquitectura UF, de acuerdo con las técnicas de esta divulgación, se aplica una descomposición poliádica canónica (CP) con convolución separable en la arquitectura UF para reemplazar la convolución 3x3 en la red troncal y/u otro bloque. Las técnicas de esta divulgación también eliminan una rama de extracción de características multiescala en un bloque de residuos, eliminan las capas de activación en un bloque de cabecera (p. ej., extracción de características de entrada), así como emplean otros ajustes de parámetros.

La Figura 19 es un diagrama en bloque que ilustra una arquitectura de red de filtro en bucle de baja complejidad de ejemplo de acuerdo con uno o más aspectos de esta divulgación. Por ejemplo, la arquitectura de la Figura 19 puede usarse para implementar un filtro en bucle de baja complejidad para el procesamiento de Y,CbCr 4:2:0

unido. El ejemplo de la Figura 19 también es aplicable para el procesamiento de componentes separados Y,CbCr. El ejemplo de la Figura 19 puede ser un filtro unificado. Un filtro unificado puede incluir un filtro que tenga al menos una entrada de calidad, tal como una entrada QP, de tal manera que el filtro pueda usarse para manejar diferentes niveles de calidad. Como puede verse en la Figura 19, el filtro de ejemplo de la Figura 19 es un filtro unificado que tiene tanto una entrada QP_{SLICE} 1902 como una entrada QP_{BASE} 1904. Un filtro unificado basado en NN, tal como el de la Figura 19, puede entrenarse usando diversos datos de imagen y/o video a diferentes niveles de calidad.

La arquitectura de la Figura 19 incluye el bloque de cabecera 1980, el bloque de transición 1922, el conjunto 1928 de uno o más bloques troncales 1930 y el bloque de cola 1986. Las entradas al bloque de cabecera 1980 incluyen IPB 1900, QP_{SLICE} 1902, QP_{BASE} 1904, BS 1906, PRED 1908 y REC 1909.

El bloque de cabecera 1980 incluye los bloques del filtro de la Figura 19 anteriores al bloque de fusión 1916. El bloque de cabecera 1980 incluye 3x3 convoluciones 1910A-1910F, que se aplican a las entradas correspondientes. Aunque se muestran como convoluciones 3x3, en algunos ejemplos, cualquiera de, o cualquier combinación de convoluciones 3x3 1910A-1910F, puede reemplazarse por una descomposición CP, como se analiza más adelante en la presente descripción. El bloque de cabecera 1980 se muestra

con las unidades de activación opcionales 1912A-1912F. En algunos ejemplos, no hay unidades de activación presentes en el bloque de cabecera 1980. En algunos ejemplos, una o más de las unidades de activación 1912A-1912F es una unidad
5 lineal rectificadora (ReLU). En algunos ejemplos, el bloque de cabecera 1980 puede incluir una o más ramas multiescala (no mostradas por motivos de simplicidad), tales como la rama multiescala 1620 de la Figura 16.

El bloque de fusión 1916 puede fusionar los mapas
10 de características usando convolución 1x1 1918 y PReLU 1920. El bloque de transición 1922 puede incluir bloques del filtro 1980 que se producen después del bloque de fusión 1916 y antes de cualquier bloque de filtro o troncal (p. ej., antes de los bloques de filtro 1930A-1930N). El bloque de
15 transición 1922 procesa los datos fusionados usando la convolución 3x3 1924 y PReLU 1926. En algunos ejemplos, la convolución 3x3 1924 puede ser reemplazada por una descomposición CP, como se analiza adicionalmente en la presente descripción. El conjunto 1928 puede incluir bloques
20 de filtro 1930A-1930N, que también pueden denominarse bloques troncales. En algunos ejemplos, los bloques de filtro 1930A-1930N pueden proporcionar funciones de filtrado del filtro de la Figura 19. En algunos ejemplos, los bloques de filtro 1930A-1930N pueden ubicarse después del bloque de transición
25 1922 y antes del bloque de cola 1986. En algunos ejemplos, uno o más de los bloques de filtro 1930A-1930N pueden incluir

una descomposición CP, como se analiza adicionalmente en la presente descripción. En algunos ejemplos, el conjunto 1928 (p. ej., bloques de filtro 1930A-1930N) no incluye una rama multiescala, tal como la rama multiescala 1620 de la Figura 16. La eliminación u omisión de la sucursal multiescala puede reducir la complejidad del filtro basado en NN.

El bloque de cola 1986 puede incluir los bloques del filtro de la Figura 19 después de todos los bloques troncales o bloques de filtro (p. ej., después de los bloques de filtro 1930A-1930N). El bloque de cola 1986 puede incluir una convolución 3x3 1950, una PReLU 1952, una convolución 3x3 1954, una unidad de recorte 1958, una unidad de mezcla de píxeles 1956 y una unidad de recorte 1960. En algunos ejemplos, uno o más de la convolución 3x3 1950 o la convolución 3x3 1954 pueden reemplazarse por una descomposición CP, como se analiza adicionalmente en la presente descripción.

Los parámetros expuestos en la presente descripción para el ejemplo de la Figura 19 son solo un ejemplo de las posibles opciones. El rendimiento de tal arquitectura puede variar según las diferentes configuraciones. En el ejemplo de la Figura 19, se describen los siguientes elementos: activación en el bloque de cabecera; rama multiescala; y la descomposición CP. Cabe señalar que todos, cualquiera o cualquier

combinación de tales elementos pueden ser utilizados de acuerdo con las técnicas de esta divulgación.

En la Figura 19 se muestra una arquitectura de ejemplo y ejemplo de los parámetros, tales como número de
5 entradas, características, canales y número de bloque de residuos. En algunos ejemplos, los parámetros para el ejemplo de la Figura 19 incluyen uno o más de: $d_1 = 12$, $d_2 = 8$, $d_3 = 4$, $d_4 = 2$, $d_5 = 2$, $C = 24$, $N = 11$, y $d_6 = 24$. La complejidad es del orden de 19,55 kMAC para la configuración ilustrada.

10 Por ejemplo, con respecto a la activación en el bloque de cabecera 1980, en algunos ejemplos, las unidades de activación 1912A-1912F, que pueden ser PReLU en la arquitectura UF actual, se eliminan para un grupo de entradas. La eliminación de PReLU puede reducir la
15 complejidad del filtro basado en NN. Por ejemplo, el bloque de cabecera 1980 puede no incluir una PReLU correspondiente para al menos una entrada al bloque de cabecera 1980. En algunos ejemplos, cualquiera de, o cualquier combinación de, unidades de activación 1912A-1912E puede retirarse del bloque
20 de cabecera 1980. Por ejemplo, el bloque de cabecera 1980 puede no incluir una capa de activación correspondiente para al menos una entrada al bloque de cabecera 1980. En algunos ejemplos, una o más de las unidades de activación 1912A-1912F pueden ser una unidad lineal rectificadora ReLU. Reemplazar
25 una PReLU por una ReLU puede reducir la complejidad del filtro basado en NN. Por ejemplo, uno o más de los PReLU en la

arquitectura UF actual pueden ser reemplazados por una ReLU. En algunos ejemplos, las unidades de activación 1912A-1912F para el bloque de cabecera 1980 se retiran para permitir que solo se realice(n) operación(es) lineal(es) antes de la
5 operación de fusión. Por ejemplo, las salidas de una o más de las convoluciones 3x3 1910A-1910F se ingresan a convolución 1x1 1918 sin pasar a través de ninguna capa de activación. La eliminación de las capas de activación puede reducir la complejidad del filtro basado en NN.

10 Por ejemplo, con respecto a las ramificaciones multiescala, en algunos ejemplos, la rama multiescala de la arquitectura UF actual (p. ej., como se muestra en la Figura 16) se elimina de los bloques de residuos de la red troncal (p. ej., los bloques de filtro 1930A-1930N) de
15 manera que los bloques troncales no realizan convoluciones paralelas. La eliminación u omisión de la sucursal multiescala puede reducir la complejidad del filtro basado en NN. En algunos ejemplos, la rama multiescala se mueve desde los bloques troncales (p. ej., una rama multiescala
20 se muestra en la Figura 16 en un bloque troncal) al bloque de cabecera 1980 para facilitar la extracción de características multiescala a un costo computacional relativamente bajo (p. ej., menor que si la rama multiescala se implementara en los bloques troncal). Por
25 ejemplo, el bloque de cabecera 1980 puede incluir al menos una rama multiescala similar a la rama multiescala 1620.

En algunos ejemplos, la rama multiescala puede aplicarse a REC 1909, p. ej., antes de la convolución 3x3 1910F.

Por ejemplo, con respecto a la descomposición CP, se puede usar una descompensación CP para descomponer 5 convoluciones multidimensionales, por ejemplo, usando 1) una convolución 1x1, y 2) pluralidad de convoluciones separadas en una dimensión horizontal y en una dimensión vertical (p. ej., una convolución 1x3 y una convolución 3x1), seguida de 3) una convolución 1x1 para combinar las características 10 extraídas de las anteriores capas. En algunos ejemplos, la primera convolución 1x1 de la descomposición CP puede combinarse con una capa previa, por lo tanto, se puede omitir la primera convolución 1x1. Por ejemplo, la descomposición CP puede incluir al menos una convolución 1x1 y convoluciones 15 3x1 separables en una dimensión horizontal y una vertical. Por ejemplo, las convoluciones separadas en la dimensión horizontal y la dimensión vertical pueden incluir una convolución en la dimensión horizontal y una convolución en la dimensión vertical. La convolución en la dimensión 20 horizontal puede incluir una de una convolución 1x3 o una convolución 3x1. La convolución en la dimensión vertical puede incluir la otra de la convolución 1x3 o la convolución 3x1. En algunos ejemplos, la al menos una convolución 1x1 incluye una primera convolución 1x1 que ocurre antes de la 25 convolución en la dimensión horizontal y la convolución en la dimensión vertical, y una segunda convolución 1x1 que

ocurre después de la convolución en la dimensión horizontal y la convolución en la dimensión vertical. En algunos ejemplos, las primeras convoluciones 1×1 pueden omitirse si no hay expansión o contracción de canales. En algunos
5 ejemplos, si el tamaño del núcleo de convolución es $N \times M$, la descomposición espacial del núcleo puede aproximarse mediante una convolución $N \times 1$ seguida de una convolución $1 \times M$.

En algunos ejemplos, el codificador de video 200 o el decodificador de video 300 pueden emplear descomposición
10 separable (p. ej., descomposición CP) en el conjunto 1928 y/o el bloque de cola 1986. En algunos ejemplos, el codificador de video 200 o el decodificador de video 300 pueden aplicar la descomposición separable (p. ej., descomposición CP) a todos los bloques (p. ej., bloque de
15 cabecera 1980, bloque de transición 1922, conjunto 1928 y bloque de cola 1986). Por ejemplo, una descomposición CP puede ser un tipo específico de descomposición separable y puede usar una pluralidad de convoluciones para aproximarse a una convolución multidimensional de orden superior. Por
20 ejemplo, una descomposición CP puede aproximarse a una convolución 3×3 usando una convolución 1×1 , una convolución 1×3 , una convolución 3×1 y otra convolución 1×1 .

La Figura 20 es un diagrama en bloque que ilustra un bloque troncal de ejemplo de acuerdo con uno o más
25 aspectos de esta divulgación. El bloque troncal 2000 puede ser un ejemplo de cualquiera de los bloques de filtro 1930A-

1930N del conjunto 1928 de la Figura 19. El bloque troncal 2000 difiere del bloque troncal 1600 de la UF en que el bloque troncal 2000 no incluye una rama multiescala como el bloque troncal 1600 y que la convolución multidimensional reemplazada por las convoluciones 1610, 1612 y 1614, están en el bloque troncal 2000 reemplazada por una descomposición CP. La descomposición CP en el bloque troncal 2000 incluye una convolución 1x1 2006, una convolución 1x3 2008, una convolución 3x1 2010, y una convolución 1x1 2012. En algunos ejemplos, los parámetros de la Figura 20 incluyen $C = 24$, $C1 = 72$ y $C21 = 24$.

La Figura 21 es un diagrama en bloque que ilustra una descomposición CP de ejemplo, de acuerdo con uno o más aspectos de esta divulgación. La descomposición CP 2100 puede incluir una convolución 1x1 2106, una convolución 1x3 2108, una convolución 3x1 2110 y una convolución 1x1 2112. Debe entenderse que la descomposición CP 2100 puede reemplazar cualquiera de, o cualquier combinación de, convoluciones 3x3 1910A-1910F del bloque de cabecera 1980, convolución 3x3 1924 del bloque de transición 1922, convolución 3x3 1950 y/o convolución 3x3 1954 del bloque de cola 1986. En algunos ejemplos, los parámetros de la Figura 21 incluyen $C = 24$, $C1 = 72$ y $C21 = 24$.

Las técnicas divulgadas en la presente pueden ser aplicables a filtros en bucle con una amplia gama de complejidad eligiendo diferentes configuraciones de

parámetros. Las técnicas de esta divulgación pueden reducir la complejidad de cálculo y los requerimientos de ancho de banda de memoria y proporcionar rendimiento competitivo, tal como para dispositivos con complejidad y/o limitaciones de ancho de banda de memoria. Los ejemplos descritos en esta divulgación están relacionados con el filtrado en bucle basado en NN, sin embargo, tales ejemplos pueden ser aplicables a cualquier herramienta de codificación de video basada en NN que consuma datos de entrada con ciertas propiedades estadísticas, tales como contenido estático o representación dispersa.

La Figura 22 es un diagrama de flujo que ilustra técnicas de filtrado en bucle basadas en NN de ejemplo de acuerdo con uno o más aspectos de esta divulgación. El decodificador de video 300 puede recibir datos para reconstruir una imagen de datos de video (2200). Por ejemplo, el codificador de video 200 puede transmitir datos de video codificados de la imagen en un flujo de bits al decodificador de video 300 y el decodificador de video 300 puede recibir los datos de video codificados de la imagen en el flujo de bits.

El decodificador de video 300 puede reconstruir la imagen de datos de video (2202). Por ejemplo, el decodificador de video 300 puede decodificar los datos de video codificados para reconstruir la imagen de datos de video.

El decodificador de video 300 puede aplicar un filtro basado en red neuronal (NN) a la imagen reconstruida de datos de video, en donde el filtro basado en NN comprende un filtro unificado que comprende un bloque de cabecera, un bloque de transición, uno o más bloques troncales y un bloque de cola, y en donde al menos uno del bloque de cabecera, el bloque de transición, el uno o más bloques troncales, o el bloque de cola comprende una descomposición poliádica canónica (CP) con convolución separable (2204). Por ejemplo, el decodificador de video 300 puede filtrar la imagen reconstruida usando un filtro basado en NN, tal como el de la Figura 19. El filtro basado en NN puede incluir el bloque de cabecera 1980, el bloque de transición 1922, uno o más bloques troncales (p. ej., el conjunto 1928) y el bloque de cola 1986. Al menos uno del bloque de cabecera 1980, bloque de transición 1922, uno o más bloques troncales, o bloque de cola 1986 incluye una descomposición CP con convolución separable. El uno o más bloques troncales pueden ser bloques residuales.

Por ejemplo, el bloque de cabecera 1980 puede incluir los bloques del filtro 19 basado en NN antes de un bloque de fusión donde se fusionan los mapas de características. El bloque de transición 1922 puede incluir bloques del filtro basado en NN que se producen después de fusionar los mapas de características (p. ej., después del bloque de fusión y, por lo tanto, después del bloque de

cabecera) y antes de cualquier bloque troncal o de filtro. Los bloques troncales (p. ej., los bloques de filtro 1930A-1930N) pueden proporcionar funciones de filtro e incluir bloques del filtro basado en NN entre el bloque de transición 1922 y el bloque de cola 1986. El bloque de cola 1986 puede incluir bloques del filtro basado en NN que se producen después de todos los bloques troncales.

En algunos ejemplos, la descomposición CP incluye al menos una convolución 1x1, una convolución en una dimensión horizontal y una convolución en una dimensión vertical. En algunos ejemplos, la convolución en la dimensión horizontal comprende una de una convolución 1x3 o una convolución 3x1, y en donde la convolución en la dimensión vertical incluye otra de la convolución 1x3 o la convolución 3x1. Por ejemplo, la descomposición CP puede incluir al menos una convolución 1x1, una convolución 1x3 y una convolución 3x1. En algunos ejemplos, la al menos una convolución 1x1 incluye una primera convolución 1x1 que ocurre antes de la convolución en la dimensión horizontal y la convolución en la dimensión vertical y una segunda convolución 1x1 que ocurre después de la convolución en la dimensión horizontal y la convolución en la dimensión vertical. Cabe señalar que el orden de las convoluciones puede ser cambiado y las dos capas lineales pueden combinarse. En algunos ejemplos, la segunda convolución 1x1 combina características extraídas de capas anteriores.

En algunos ejemplos, uno o más bloques troncales y el bloque de cola 1986 incluyen cada uno una descomposición CP correspondiente con convolución separable. En otras palabras, el uno o más bloques troncales incluye una descomposición CP con convolución separable y el bloque de cola incluye una descomposición CP con convolución separable. En algunos ejemplos, el bloque de cabecera 1980, el bloque de transición 1922, uno o más bloques troncales 1930 y el bloque de cola 1986 comprenden cada uno una descomposición CP correspondiente con convolución separable. En algunos ejemplos, uno o más bloques troncales 1930 no incluyen una rama multiescala. Por ejemplo, una rama multiescala puede realizar extracción de características multiescala mediante el uso de convoluciones paralelas tal como se muestra en la Figura 16.

En algunos ejemplos, el bloque de cabecera 1980 no incluye una correspondiente activación de unidad lineal rectificadora paramétrica (PReLU) para al menos una entrada al bloque de cabecera 1980. En algunos ejemplos, el bloque de cabecera 1980 no incluye una capa de activación correspondiente para al menos una entrada al bloque de cabecera. En algunos ejemplos, el bloque de cabecera 1980 no incluye ninguna capa de activación. En algunos ejemplos, el bloque de cabecera 1980 incluye al menos una activación de unidad lineal rectificadora (ReLU) correspondiente para al menos una entrada al bloque de cabecera 1980. En algunos

ejemplos, el bloque de cabecera 1980 incluye al menos una rama multiescala. En algunos ejemplos, el uno o más bloques troncales del filtro basado en NN incluye cada uno 24 canales. Por ejemplo, C=24.

5 La Figura 23 es un diagrama en bloque que ilustra un ejemplo de codificador de video 200 que puede realizar las técnicas de esta divulgación. La Figura 23 se proporciona con fines de explicación y no debe considerarse limitante de las técnicas como se ejemplifica y describe
10 ampliamente en esta divulgación. Para fines de explicación, esta descripción divulga el codificador de video 200 de acuerdo con las técnicas de VVC y HEVC. Sin embargo, las técnicas de esta divulgación pueden ser realizadas por dispositivos de codificación de video que
15 se configuran para otros estándares de codificación de video y formatos de codificación de video, tales como AV1 y sucesores del formato de codificación de video AV1.

 En el ejemplo de la Figura 23, el codificador de video 200 incluye la memoria de datos de video 230, la
20 unidad de selección de modo 202, la unidad de generación residual 204, la unidad de procesamiento de transformación 206, la unidad de cuantificación 208, la unidad de cuantificación inversa 210, la unidad de procesamiento de transformación inversa 212, la unidad de reconstrucción
25 214, la unidad de filtro 216, el búfer de imágenes decodificadas (DPB, por sus siglas en inglés) 218 y la

unidad de codificación por entropía 220. Cualquiera o la totalidad de la memoria de datos de video 230, la unidad de selección de modo 202, la unidad de generación residual 204, la unidad de procesamiento de transformación 206, la unidad de cuantificación 208, la unidad de cuantificación inversa 210, la unidad de procesamiento de transformación inversa 212, la unidad de reconstrucción 214, la unidad de filtro 216, DPB 218 y la unidad de codificación por entropía 220 pueden implementarse en uno o más procesadores o en circuitos de procesamiento. Por ejemplo, las unidades del codificador de video 200 pueden implementarse como uno o más circuitos o elementos lógicos como parte de circuitos de hardware, o como parte de un procesador, ASIC o FPGA. Además, el codificador de video 200 puede incluir procesadores o circuitos de procesamiento adicionales o alternativos para realizar estas y otras funciones.

La memoria de datos de video 230 puede almacenar datos de video para codificarse por los componentes del codificador de video 200. El codificador de video 200 puede recibir los datos de video almacenados en la memoria de datos de video 230 desde, por ejemplo, la fuente de video 104 (Figura 1). El DPB 218 puede actuar como una memoria de imagen de referencia que almacena datos de video de referencia para su uso en la predicción de datos de video subsiguientes por el codificador de video 200. La memoria de datos de video 230 y el DPB 218 pueden estar formadas por cualquiera de una

variedad de dispositivos de memoria, tales como memoria de acceso aleatorio dinámico (DRAM, por sus siglas en inglés), incluidos DRAM síncrona (SDRAM, por sus siglas en inglés), RAM magnetorresistiva (MRAM, por sus siglas en inglés), RAM resistiva (RRAM, por sus siglas en inglés) u otros tipos de dispositivos de memoria. La memoria de datos de video 230 y DPB 218 pueden proporcionarse mediante el mismo dispositivo de memoria o dispositivos de memoria separados. En diversos ejemplos, la memoria de datos de video 230 puede estar en chip con otros componentes del codificador de video 200, como se ilustra, o fuera del chip en relación con esos componentes.

En esta descripción, la referencia a la memoria de datos de video 230 no debe interpretarse como limitada a la memoria interna del codificador de video 200, a menos que se describan específicamente como tal, o a la memoria externa al codificador de video 200, a menos que se divulguen específicamente como tal. Más bien, la referencia a la memoria de datos de video 230 debe entenderse como una memoria de referencia que almacena datos de video que el codificador de video 200 recibe para su codificación (p. ej., datos de video para un bloque actual que se codificará). La memoria 106 de la Figura 1 también puede proporcionar almacenamiento temporal de salidas de las diversas unidades del codificador de video 200.

Las diversas unidades de la Figura 23 se ilustran para ayudar a comprender las operaciones realizadas por el

codificador de video 200. Las unidades pueden implementarse como circuitos de función fija, circuitos programables, o una combinación de estos. Los circuitos de función fija se refieren a circuitos que proporcionan una funcionalidad particular y están preestablecidos en las operaciones que se pueden realizar. Los circuitos programables se refieren a circuitos que se pueden programar para realizar diversas tareas, y proporcionan una funcionalidad flexible en las operaciones que se pueden realizar. Por ejemplo, los circuitos programables pueden ejecutar software o firmware que hacen que los circuitos programables funcionen de la manera definida por las instrucciones del software o firmware. Los circuitos de función fija pueden ejecutar instrucciones de software (p. ej., para recibir parámetros o parámetros de salida), pero los tipos de operaciones que realizan los circuitos de función fija son generalmente inmutables. En algunos ejemplos, una o más de las unidades pueden ser bloques de circuito distintos (función fija o programable), y en algunos ejemplos, una o más de las unidades pueden ser circuitos integrados.

El codificador de video 200 puede incluir unidades lógicas aritméticas (ALU, por sus siglas en inglés), unidades de función elemental (EFU, por sus siglas en inglés), circuitos digitales, circuitos analógicos y/o núcleos programables, formados a partir de circuitos programables. En ejemplos en los que las operaciones del

codificador de video 200 se realizan usando software
ejecutado por los circuitos programables, la memoria 106
(Figura 1) puede almacenar las instrucciones (p. ej.,
código objeto) del software que recibe y ejecuta el
5 codificador de video 200, u otra memoria dentro del
codificador de video 200 (no mostrado) puede almacenar
dichas instrucciones.

La memoria de datos de video 230 se configura
para almacenar datos de video recibidos. El codificador de
10 video 200 puede recuperar una imagen de los datos de video
de la memoria de datos de video 230 y proporcionar los
datos de video a la unidad de generación residual 204 y a
la unidad de selección de modo 202. Los datos de video en
la memoria de datos de video 230 pueden ser datos de video
15 sin procesar a codificarse.

La unidad de selección de modo 202 incluye una
unidad de estimación de movimiento 222, una unidad de
compensación de movimiento 224 y una unidad de
intrapredicción 226. La unidad de selección de modo 202
20 puede incluir unidades funcionales adicionales para
realizar predicción de video de acuerdo con otros modos de
predicción. Como ejemplos, la unidad de selección de modo
202 puede incluir una unidad de paleta, una unidad de copia
intrabloque (que puede ser parte de la unidad de estimación
25 de movimiento 222 y/o la unidad de compensación de

movimiento 224), una unidad afín, una unidad de modelo lineal (LM, por sus siglas en inglés), o lo similar.

La unidad de selección de modo 202 coordina generalmente múltiples pasadas de codificación para probar
5 combinaciones de parámetros de codificación y valores de tasa de distorsión resultantes para dichas combinaciones. Los parámetros de codificación pueden incluir la partición de las CTU en CU, modos de predicción para las CU, tipos de transformación para datos residuales de las CU, parámetros
10 de cuantificación para datos residuales de las CU, y así sucesivamente. La unidad de selección de modo 202 puede seleccionar en última instancia la combinación de parámetros de codificación que tienen valores de tasa de distorsión que son mejores que las otras combinaciones probadas.

15 El codificador de video 200 puede particionar una imagen recuperada de la memoria de datos de video 230 en una serie de CTU, y encapsular una o más CTU dentro de una porción. La unidad de selección de modo 202 puede particionar una CTU de la imagen de acuerdo con una
20 estructura de árbol, tal como la estructura MTT, estructura QTBT, estructura de superbloque, o la estructura de árbol cuaternario descrita anteriormente. Como se describió anteriormente, el codificador de video 200 puede formar una o más CU a partir de la partición de una CTU de acuerdo
25 con la estructura de árbol. Tal CU también puede denominarse generalmente “bloque de video” o “bloque”.

En general, la unidad de selección de modo 202 también controla los componentes de esta (p. ej., la unidad de estimación de movimiento 222, la unidad de compensación de movimiento 224 y la unidad de intrapredicción 226) para
5 generar un bloque de predicción para un bloque actual (p. ej., una CU actual, o en HEVC, la porción superpuesta de una PU y una TU). Para la interpredicción de un bloque actual, la unidad de estimación de movimiento 222 puede realizar una búsqueda de movimiento para identificar uno o
10 más bloques de referencia que coinciden estrechamente en una o más imágenes de referencia (p. ej., una o más imágenes codificadas previamente almacenadas en DPB 218). En particular, la unidad de estimación de movimiento 222 puede calcular un valor representativo de cuán similar es un
15 bloque de referencia potencial al bloque actual, p. ej., de acuerdo con la suma de la diferencia absoluta (SAD), la suma de diferencias cuadradas (SSD), la diferencia absoluta media (MAD), las diferencias cuadradas medias (MSD), o lo similar. La unidad de estimación de movimiento 222
20 generalmente puede realizar estos cálculos usando diferencias muestra por muestra entre el bloque actual y el bloque de referencia que se está considerando. La unidad de estimación de movimiento 222 puede identificar un bloque de referencia que tiene un valor más bajo resultante de
25 estos cálculos, indicando un bloque de referencia que coincide más estrechamente con el bloque actual.

La unidad de estimación de movimiento 222 puede formar uno o más vectores de movimiento (MV, por sus siglas en inglés) que definen las posiciones de los bloques de referencia en las imágenes de referencia en relación con la posición del bloque actual en una imagen actual. La unidad de estimación de movimiento 222 puede proporcionar entonces los vectores de movimiento a la unidad de compensación de movimiento 224. Por ejemplo, para la interpredicción unidireccional, la unidad de estimación de movimiento 222 puede proporcionar un único vector de movimiento, mientras que para la interpredicción bidireccional, la unidad de estimación de movimiento 222 puede proporcionar dos vectores de movimiento. La unidad de compensación de movimiento 224 puede generar entonces un bloque de predicción usando los vectores de movimiento. Por ejemplo, la unidad de compensación de movimiento 224 puede recuperar datos del bloque de referencia usando el vector de movimiento. Como otro ejemplo, si el vector de movimiento tiene una precisión de muestra fraccional, la unidad de compensación de movimiento 224 puede interpolar valores para el bloque de predicción de acuerdo con uno o más filtros de interpolación. Además, para la interpredicción bidireccional, la unidad de compensación de movimiento 224 puede recuperar datos para dos bloques de referencia identificados por vectores de movimiento respectivos y combinar los datos recuperados, p. ej., a

través del promedio muestra por muestra o el promedio ponderado.

Cuando se opera de acuerdo con el formato de codificación de video AV1, la unidad de estimación de movimiento 222 y la unidad de compensación de movimiento 224 pueden configurarse para codificar bloques de codificación de datos de video (p. ej., bloques de codificación de luma y croma) usando compensación de movimiento traslacional, compensación de movimiento afín, compensación de movimiento en bloque superpuesto (OBMC) y/o predicción inter-intracompuesta.

Como otro ejemplo, para la intrapredicción, o codificación intrapredicción, la unidad de intrapredicción 226 puede generar el bloque de predicción a partir de muestras vecinas al bloque actual. Por ejemplo, para modos direccionales, la unidad de intrapredicción 226 puede generalmente combinar matemáticamente valores de muestras vecinas y rellenar estos valores calculados en la dirección definida a través del bloque actual para producir el bloque de predicción. Como otro ejemplo, para el modo DC, la unidad de intrapredicción 226 puede calcular un promedio de las muestras vecinas al bloque actual y generar el bloque de predicción para incluir este promedio resultante para cada muestra del bloque de predicción.

Cuando se opera de acuerdo con el formato de codificación de video AV1, la unidad de intrapredicción 226

puede configurarse para codificar bloques de codificación de datos de video (p. ej., bloques de codificación tanto de luma como de croma) usando predicción intradireccional, intrapredicción no direccional, intrapredicción de filtro
5 recursivo, predicción croma-de-luma (CFL, por sus siglas en inglés), copia intrabloque (IBC, por sus siglas en inglés) y/o modo de paleta de colores. La unidad de selección de modo 202 puede incluir unidades funcionales adicionales para realizar predicción de video de acuerdo
10 con otros modos de predicción.

La unidad de selección de modo 202 proporciona el bloque de predicción a la unidad de generación residual 204. La unidad de generación residual 204 recibe una versión sin codificar del bloque actual desde la memoria de datos de
15 video 230 y el bloque de predicción de la unidad de selección de modo 202. La unidad de generación residual 204 calcula las diferencias muestra por muestra entre el bloque actual y el bloque de predicción. Las diferencias resultantes, muestra por muestra, definen un bloque residual para el bloque
20 actual. En algunos ejemplos, la unidad de generación residual 204 también puede determinar diferencias entre los valores de muestra en el bloque residual para generar un bloque residual usando modulación de código de pulso diferencial residual (RDPCM, por sus siglas en inglés). En algunos
25 ejemplos, la unidad de generación residual 204 puede formarse

usando uno o más circuitos sustractores que realizan sustracción binaria.

En ejemplos donde la unidad de selección de modo 202 divide las CU en PU, cada PU puede estar asociada con una unidad de predicción de luma y unidades de predicción de cromas correspondientes. El codificador de video 200 y el decodificador de video 300 pueden soportar PU que tienen varios tamaños. Como se indicó anteriormente, el tamaño de una CU puede referirse al tamaño del bloque de codificación de luma de la CU y el tamaño de una PU puede referirse al tamaño de una unidad de predicción de luma de la PU. Suponiendo que el tamaño de una CU particular es $2N \times 2N$, el codificador de video 200 puede soportar tamaños de PU de $2N \times 2N$ o $N \times N$ para la intrapredicción, y tamaños de PU simétricos de $2N \times 2N$, $2N \times N$, $N \times 2N$, $N \times N$ o similares para la interpredicción. El codificador de video 200 y el decodificador de video 300 también pueden soportar particiones asimétricas para tamaños de PU de $2N \times nU$, $2N \times nD$, $nL \times 2N$ y $nR \times 2N$ para la interpredicción.

En ejemplos en los que la unidad de selección de modo 202 no divide adicionalmente una CU en PU, cada CU puede estar asociada con un bloque de codificación de luma y bloques de codificación de cromas correspondientes. Como anteriormente, el tamaño de una CU puede referirse al tamaño del bloque de codificación de luma de la CU. El

codificador de video 200 y el decodificador de video 300 pueden soportar tamaños de CU de $2N \times 2N$, $2N \times N$ o $N \times 2N$.

Para otras técnicas de codificación de video tales como una codificación en modo de copia intrabloque, una codificación en modo afín y codificación en modo de modelo lineal (LM), como algunos ejemplos, la unidad de selección de modo 202, a través de unidades respectivas asociadas con las técnicas de codificación, genera un bloque de predicción para el bloque actual que se está codificando. En algunos ejemplos, como la codificación en modo paleta, la unidad de selección de modo 202 puede no generar un bloque de predicción, y en su lugar generar elementos de sintaxis que indican la manera en la que se reconstruye el bloque basado en una paleta seleccionada. En tales modos, la unidad de selección de modo 202 puede proporcionar estos elementos de sintaxis a la unidad de codificación por entropía 220 a ser codificada.

Como se describió anteriormente, la unidad de generación residual 204 recibe los datos de video para el bloque actual y el bloque de predicción correspondiente. La unidad de generación residual 204 genera entonces un bloque residual para el bloque actual. Para generar el bloque residual, la unidad de generación residual 204 calcula las diferencias muestra por muestra entre el bloque de predicción y el bloque actual.

La unidad de procesamiento de transformación 206 aplica una o más transformaciones al bloque residual para generar un bloque de coeficientes de transformación (denominado en la presente “bloque de coeficiente de transformación”). La unidad de procesamiento de transformación 206 puede aplicar diversas transformaciones a un bloque residual para formar el bloque de coeficiente de transformación. Por ejemplo, la unidad de procesamiento de transformación 206 puede aplicar una transformación discreta de coseno (DCT), una transformación direccional, una transformación Karhunen-Loeve (KLT) o una transformación conceptualmente similar a un bloque residual. En algunos ejemplos, la unidad de procesamiento de transformación 206 puede realizar múltiples transformaciones a un bloque residual, p. ej., una transformación primaria y una transformación secundaria, tal como una transformación rotacional. En algunos ejemplos, la unidad de procesamiento de transformación 206 no aplica transformaciones a un bloque residual.

Cuando se opera de acuerdo con AV1, la unidad de procesamiento de transformación 206 puede aplicar una o más transformaciones al bloque residual para generar un bloque de coeficientes de transformación (denominado en la presente “bloque de coeficiente de transformación”). La unidad de procesamiento de transformación 206 puede aplicar diversas transformaciones a un bloque residual para formar el bloque

de coeficiente de transformación. Por ejemplo, la unidad de procesamiento de transformación 206 puede aplicar una combinación de transformación horizontal/vertical que puede incluir una transformación discreta de coseno (DCT), una
5 transformación discreta de seno asimétrica (ADST, por sus siglas en inglés), un ADST volteado (p. ej., un ADST en orden inverso) y una transformación de identidad (IDTX, por sus siglas en inglés). Cuando se usa una transformación de identidad, la transformación se omite en una de las
10 direcciones vertical u horizontal. En algunos ejemplos, se puede omitir el procesamiento de transformación.

La unidad de cuantificación 208 puede cuantificar los coeficientes de transformación en un bloque de coeficientes de transformación, para producir un
15 bloque de coeficiente de transformación cuantificado. La unidad de cuantificación 208 puede cuantificar coeficientes de transformación de un bloque de coeficientes de transformación de acuerdo con un valor de parámetro de cuantificación (QP, por sus siglas en inglés)
20 asociado con el bloque actual. El codificador de video 200 (p. ej., a través de la unidad de selección de modo 202) puede ajustar el grado de cuantificación aplicado a los bloques de coeficientes de transformación asociados con el bloque actual al ajustar el valor de QP asociado con la
25 CU. La cuantificación puede introducir pérdida de información y, por lo tanto, los coeficientes de

transformación cuantificados pueden tener menor precisión que los coeficientes de transformación originales producidos por la unidad de procesamiento de transformación 206.

5 La unidad de cuantificación inversa 210 y la unidad de procesamiento de transformación inversa 212 pueden aplicar cuantificación inversa y transformaciones inversas a un bloque de coeficientes de transformación cuantificado, respectivamente, para reconstruir un bloque residual a
10 partir del bloque de coeficientes de transformación. La unidad de reconstrucción 214 puede producir un bloque reconstruido correspondiente al bloque actual (aunque potencialmente con algún grado de distorsión) basado en el bloque residual reconstruido y un bloque de predicción
15 generado por la unidad de selección de modo 202. Por ejemplo, la unidad de reconstrucción 214 puede añadir muestras del bloque residual reconstruido a muestras correspondientes del bloque de predicción generado por la unidad de selección de modo 202 para producir el bloque reconstruido.

20 La unidad de filtro 216 puede realizar una o más operaciones de filtro en bloques reconstruidos. Por ejemplo, la unidad de filtro 216 puede realizar operaciones de desbloqueo para reducir los artefactos de fragmentación en bloques a lo largo de los bordes de las CU. Las
25 operaciones de la unidad de filtro 216 pueden omitirse, en algunos ejemplos.

Cuando se opera de acuerdo con AV1, la unidad de filtro 216 puede realizar una o más operaciones de filtro en bloques reconstruidos. Por ejemplo, la unidad de filtro 216 puede realizar operaciones de desbloqueo para reducir los artefactos de fragmentación en bloques a lo largo de los bordes de las CU. En otros ejemplos, la unidad de filtro 216 puede aplicar un filtro de mejora direccional restringido (CDEF, por sus siglas en inglés), que puede aplicarse después del desbloqueo, y puede incluir la aplicación de filtros direccionales de paso bajo no separables, no lineales, basados en direcciones de borde estimadas. La unidad de filtro 216 también puede incluir un filtro de restauración de bucle, que se aplica después de CDEF, y puede incluir un filtro Wiener normalizado simétrico separable o un filtro autoguiado dual.

El codificador de video 200 almacena bloques reconstruidos en DPB 218. Por ejemplo, en ejemplos en los que no se realizan operaciones de la unidad de filtro 216, la unidad de reconstrucción 214 puede almacenar bloques reconstruidos en DPB 218. En ejemplos donde se realizan operaciones de la unidad de filtro 216, la unidad de filtro 216 puede almacenar los bloques reconstruidos filtrados en DPB 218. La unidad de estimación de movimiento 222 y la unidad de compensación de movimiento 224 pueden recuperar una imagen de referencia de DPB 218, formada a partir de los bloques reconstruidos (y potencialmente filtrados), para

interpretar bloques de imágenes codificadas posteriormente. Además, la unidad de intrapredicción 226 puede usar bloques reconstruidos en DPB 218 de una imagen actual para intrapredicar otros bloques en la imagen actual.

5 En general, la unidad de codificación por entropía 220 puede codificar por entropía elementos de sintaxis recibidos de otros componentes funcionales del codificador de video 200. Por ejemplo, la unidad de codificación por entropía 220 puede codificar por entropía
10 bloques de coeficientes de transformación cuantificados a partir de la unidad de cuantificación 208. Como otro ejemplo, la unidad de codificación por entropía 220 puede codificar por entropía elementos de sintaxis de predicción (p. ej., información de movimiento para la interpredicción
15 o información intramodo para intrapredicción) a partir de la unidad de selección de modo 202. La unidad de codificación por entropía 220 puede realizar una o más operaciones de codificación de entropía en los elementos de sintaxis, que son otro ejemplo de datos de video, para
20 generar datos codificados por entropía. Por ejemplo, la unidad de codificación por entropía 220 puede realizar una operación de codificación de longitud variable adaptativa al contexto (CAVLC, por sus siglas en inglés), una operación CABAC, una operación de codificación de longitud variable
25 a variable (V2V, por sus siglas en inglés), una operación de codificación aritmética binaria adaptativa al contexto

basada en sintaxis (SBAC, por sus siglas en inglés), una
operación de codificación de entropía de partición de
intervalo de probabilidad (PIPE, por sus siglas en inglés),
una operación de codificación exponencial-Golomb u otro
5 tipo de entropía operación de codificación en los datos.
En algunos ejemplos, la unidad de codificación por entropía
220 puede operar en modo de derivación donde los elementos
de sintaxis no se codifican por entropía.

El codificador de video 200 puede emitir un flujo
10 de bits que incluye los elementos de sintaxis codificados
por entropía necesarios para reconstruir bloques de una
porción o imagen. En particular, la unidad de codificación
por entropía 220 puede emitir el flujo de bits.

De acuerdo con AV1, la unidad de codificación
15 por entropía 220 puede configurarse como un codificador
aritmético multisímbolo adaptativo de símbolo a símbolo.
Un elemento de sintaxis en AV1 incluye un alfabeto de N
elementos, y un contexto (p. ej., modelo de probabilidad)
incluye un conjunto de N probabilidades. La unidad de
20 codificación por entropía 220 puede almacenar las
probabilidades como funciones de distribución acumulativa
(CDF, por sus siglas en inglés) de n bits (p. ej., 15
bits). La unidad de codificación por entropía 220 puede
realizar escalado recursivo, con un factor de
25 actualización basado en el tamaño del alfabeto, para
actualizar los contextos.

Las operaciones descritas anteriormente se divulgan con respecto a un bloque. Dicha descripción debe entenderse como operaciones para un bloque de codificación de luma y/o bloques de codificación de croma. Como se describió anteriormente, en algunos ejemplos, el bloque de codificación de luma y los bloques de codificación de croma son componentes de luma y croma de una CU. En algunos ejemplos, el bloque de codificación de luma y los bloques de codificación de croma son componentes de luma y croma de una PU.

En algunos ejemplos, las operaciones realizadas con respecto a un bloque de codificación de luma no necesitan repetirse para los bloques de codificación de croma. Como ejemplo, las operaciones para identificar un vector de movimiento (MV) y una imagen de referencia para un bloque de codificación de luma no necesitan repetirse para identificar un MV y una imagen de referencia para los bloques de croma. Más bien, el MV para el bloque de codificación de luma puede escalararse para determinar el MV para los bloques de croma, y la imagen de referencia puede ser la misma. Como otro ejemplo, el proceso de intrapredicción puede ser el mismo para el bloque de codificación de luma y los bloques de codificación de croma.

El codificador de video 200 representa un ejemplo de un dispositivo configurado para codificar datos de video que incluye una o más memorias configuradas para almacenar

datos de video, y uno o más procesadores configurados para recibir una imagen de datos de video, reconstruir la imagen de datos de video y aplicar un filtro basado en red neuronal (NN) a la imagen reconstruida de datos de video, en donde el filtro basado en NN comprende un filtro unificado que comprende un bloque de cabecera, un bloque de transición, uno o más bloques troncales y un bloque de cola, y en donde al menos uno del bloque de cabecera, el bloque de transición, el uno o más bloques troncales, o el bloque de cola comprende una descomposición poliádica canónica (CP) con convolución separable.

La Figura 24 es un diagrama en bloque que ilustra un ejemplo de decodificador de video 300 que puede realizar las técnicas de esta divulgación. La Figura 24 se proporciona con fines de explicación y no es limitante en las técnicas como se ejemplifica y describe ampliamente en esta divulgación. Para fines de explicación, esta divulgación describe el decodificador de video 300 de acuerdo con las técnicas de VVC y HEVC. Sin embargo, las técnicas de esta descripción pueden ser realizadas por dispositivos de codificación de video que se configuran para otros estándares de codificación de video.

En el ejemplo de la Figura 24, el decodificador de video 300 incluye la memoria de búfer de imágenes codificadas (CPB, por sus siglas en inglés) 320, la unidad de decodificación por entropía 302, la unidad de procesamiento

de predicción 304, la unidad de cuantificación inversa 306, la unidad de procesamiento de transformación inversa 308, la unidad de reconstrucción 310, la unidad de filtro 312 y el DPB 314. Cualquiera o la totalidad de la memoria de CPB 320, la unidad de decodificación por entropía 302, la unidad de procesamiento de predicción 304, la unidad de cuantificación inversa 306, la unidad de procesamiento de transformación inversa 308, la unidad de reconstrucción 310, la unidad de filtro 312 y DPB 314 pueden implementarse en uno o más procesadores o circuitos de procesamiento. Por ejemplo, las unidades del decodificador de video 300 pueden implementarse como uno o más circuitos o elementos lógicos como parte de circuitos de hardware, o como parte de un procesador, ASIC o FPGA. Además, el decodificador de video 300 puede incluir procesadores o circuitos de procesamiento adicionales o alternativos para realizar estas y otras funciones.

La unidad de procesamiento de predicción 304 incluye la unidad de compensación de movimiento 316 y la unidad de intrapredicción 318. La unidad de procesamiento de predicción 304 puede incluir unidades adicionales para realizar la predicción de acuerdo con otros modos de predicción. Como ejemplos, la unidad de procesamiento de predicción 304 puede incluir una unidad de paleta, una unidad de copia intrabloque (que puede formar parte de la unidad de compensación de movimiento 316), una unidad afín, una unidad de modelo lineal (LM, por sus siglas en inglés),

o lo similar. En otros ejemplos, el decodificador de video 300 puede incluir más, menos o diferentes componentes funcionales.

Cuando se opera de acuerdo con AV1, la unidad de 5 compensación de movimiento 316 puede configurarse para decodificar bloques de codificación de datos de video (p. ej., bloques de codificación de luma y croma) usando compensación de movimiento traslacional, compensación de movimiento afín, OBMC y/o inter-intrapredicción compuesta, 10 como se describió anteriormente. La unidad de intrapredicción 318 puede configurarse para decodificar bloques de codificación de datos de video (p. ej., bloques de codificación tanto de luma como de croma) usando predicción intradireccional, intrapredicción no direccional, 15 intrapredicción de filtro recursivo, CFL, IBC y/o modo de paleta de colores, como se describió anteriormente.

La memoria de CPB 320 puede almacenar datos de video, tales como un flujo de bits de video codificado, para decodificarse por los componentes del decodificador de video 20 300. Los datos de video almacenados en la memoria de CPB 320 pueden obtenerse, por ejemplo, a partir del medio legible por computador 110 (figura 1). La memoria de CPB 320 puede incluir un CPB que almacena datos de video codificados (p. ej., elementos de sintaxis) de un flujo de bits de video 25 codificado. Además, la memoria de CPB 320 puede almacenar datos de video distintos de los elementos de sintaxis de una

imagen codificada, tales como datos temporales que representan salidas de las diversas unidades del decodificador de video 300. La DPB 314 almacena generalmente imágenes decodificadas, que el decodificador de video 300 puede emitir y/o usar como datos de video de referencia cuando decodifica datos o imágenes subsiguientes del flujo de bits de video codificado. La memoria de CPB 320 y DPB 314 pueden estar formadas por cualquiera de una variedad de dispositivos de memoria, tales como DRAM, incluidos SDRAM, MRAM, RRAM u otros tipos de dispositivos de memoria. La memoria de CPB 320 y DPB 314 pueden proporcionarse mediante el mismo dispositivo de memoria o dispositivos de memoria separados. En diversos ejemplos, la memoria de CPB 320 puede estar en chip con otros componentes del decodificador de video 300, o fuera del chip en relación con esos componentes.

Adicional o alternativamente, en algunos ejemplos, el decodificador de video 300 puede recuperar datos de video codificados de la memoria 120 (Figura 1). Es decir, la memoria 120 puede almacenar datos como se ha descrito anteriormente con la memoria de CPB 320. De la misma manera, la memoria 120 puede almacenar instrucciones para ser ejecutadas por el decodificador de video 300, cuando parte o la totalidad de la funcionalidad del decodificador de video 300 se implementa en software para ser ejecutado por circuitos de procesamiento del decodificador de video 300.

Las diversas unidades mostradas en la Figura 24 se ilustran para ayudar a comprender las operaciones realizadas por el decodificador de video 300. Las unidades pueden implementarse como circuitos de función fija, circuitos programables, o una combinación de estos. De manera similar a la Figura 23, los circuitos de función fija se refieren a circuitos que proporcionan una funcionalidad particular, y están preestablecidos en las operaciones que se pueden realizar. Los circuitos programables se refieren a circuitos que se pueden programar para realizar diversas tareas, y proporcionan una funcionalidad flexible en las operaciones que se pueden realizar. Por ejemplo, los circuitos programables pueden ejecutar software o firmware que hacen que los circuitos programables funcionen de la manera definida por las instrucciones del software o firmware. Los circuitos de función fija pueden ejecutar instrucciones de software (p. ej., para recibir parámetros o parámetros de salida), pero los tipos de operaciones que realizan los circuitos de función fija son generalmente inmutables. En algunos ejemplos, una o más de las unidades pueden ser bloques de circuito distintos (función fija o programable), y en algunos ejemplos, una o más de las unidades pueden ser circuitos integrados.

El decodificador de video 300 puede incluir ALU, EFU, circuitos digitales, circuitos analógicos y/o núcleos programables formados a partir de circuitos programables.

En ejemplos en los que las operaciones del decodificador de video 300 se realizan mediante software que se ejecuta en los circuitos programables, la memoria en chip o fuera del chip puede almacenar instrucciones (p. ej., código
5 objeto) del software que el decodificador de video 300 recibe y ejecuta.

La unidad de decodificación por entropía 302 puede recibir datos de video codificados desde el CPB y decodificar por entropía los datos de video para reproducir
10 elementos de sintaxis. La unidad de procesamiento de predicción 304, la unidad de cuantificación inversa 306, la unidad de procesamiento de transformación inversa 308, la unidad de reconstrucción 310 y la unidad de filtro 312 pueden generar datos de video decodificados basados en los
15 elementos de sintaxis extraídos del flujo de bits.

En general, el decodificador de video 300 reconstruye una imagen bloque por bloque. El decodificador de video 300 puede realizar una operación de reconstrucción en cada bloque individualmente (donde el bloque que se
20 está reconstruyendo actualmente, es decir, decodificado, puede denominarse “bloque actual”).

La unidad de decodificación por entropía 302 puede decodificar por entropía elementos de sintaxis que definen coeficientes de transformación cuantificados de un bloque de
25 coeficientes de transformación cuantificados, así como información de transformación, tal como un parámetro de

cuantificación (QP) y/o indicaciones de modo de transformación. La unidad de cuantificación inversa 306 puede usar el QP asociado con el bloque de coeficientes de transformación cuantificados para determinar un grado de
5 cuantificación y, de la misma manera, un grado de cuantificación inversa para aplicar la unidad de cuantificación inversa 306. La unidad de cuantificación inversa 306 puede, por ejemplo, realizar una operación de desplazamiento a la izquierda bit a bit para cuantificar de
10 manera inversa los coeficientes de transformación cuantificados. La unidad de cuantificación inversa 306 puede formar de esta manera un bloque de coeficientes de transformación que incluye coeficientes de transformación.

Después de que la unidad de cuantificación
15 inversa 306 forme el bloque de coeficiente de transformación, la unidad de procesamiento de transformación inversa 308 puede aplicar una o más transformaciones inversas al bloque de coeficientes de transformación para generar un bloque residual asociado
20 con el bloque actual. Por ejemplo, la unidad de procesamiento de transformación inversa 308 puede aplicar una DCT inversa, una transformación de enteros inversa, una transformación inversa de Karhunen-Loeve (KLT), una transformación rotacional inversa, una transformación
25 direccional inversa u otra transformación inversa al bloque de coeficientes de transformación.

Además, la unidad de procesamiento de predicción 304 genera un bloque de predicción de acuerdo con elementos de sintaxis de información de predicción que fueron decodificados por entropía por la unidad de decodificación por entropía 302. Por ejemplo, si los elementos de sintaxis de información de predicción indican que el bloque actual está interpredicho, la unidad de compensación de movimiento 316 puede generar el bloque de predicción. En este caso, los elementos de sintaxis de información de predicción pueden indicar una imagen de referencia en DPB 314 desde la cual recuperar un bloque de referencia, así como un vector de movimiento que identifica una ubicación del bloque de referencia en la imagen de referencia en relación con la ubicación del bloque actual en la imagen actual. La unidad de compensación de movimiento 316 generalmente puede realizar el proceso de interpredicción de una manera que es sustancialmente similar a la descrita con respecto a la unidad de compensación de movimiento 224 (Figura 23).

Como otro ejemplo, si los elementos de sintaxis de información de predicción indican que el bloque actual está intrapredicho, la unidad de intrapredicción 318 puede generar el bloque de predicción de acuerdo con un modo de intrapredicción indicado por los elementos de sintaxis de información de predicción. De nuevo, la unidad de intrapredicción 318 puede realizar generalmente el proceso de intrapredicción de una manera que es sustancialmente

similar a la descrita con respecto a la unidad de intrapredicción 226 (Figura 23). La unidad de intrapredicción 318 puede recuperar datos de muestras vecinas al bloque actual de DPB 314.

5 La unidad de reconstrucción 310 puede reconstruir el bloque actual usando el bloque de predicción y el bloque residual. Por ejemplo, la unidad de reconstrucción 310 puede añadir muestras del bloque residual a muestras correspondientes del bloque de
10 predicción para reconstruir el bloque actual.

 La unidad de filtro 312 puede realizar una o más operaciones de filtro en bloques reconstruidos. Por ejemplo, la unidad de filtro 312 puede realizar operaciones de desbloqueo para reducir los artefactos de fragmentación
15 en bloques a lo largo de los bordes de los bloques reconstruidos. Las operaciones de la unidad de filtro 312 no se realizan necesariamente en todos los ejemplos.

 El decodificador de video 300 puede almacenar los bloques reconstruidos en DPB 314. Por ejemplo, en ejemplos
20 en los que no se realizan operaciones de la unidad de filtro 312, la unidad de reconstrucción 310 puede almacenar bloques reconstruidos en DPB 314. En ejemplos donde se realizan operaciones de la unidad de filtro 312, la unidad de filtro 312 puede almacenar los bloques reconstruidos filtrados en
25 DPB 314. Como se discutió anteriormente, DPB 314 puede proporcionar información de referencia, tal como muestras de

una imagen actual para intrapredicción e imágenes previamente decodificadas para compensación de movimiento posterior, a la unidad de procesamiento de predicción 304. Además, el decodificador de video 300 puede emitir imágenes decodificadas (p. ej., video decodificado) desde DPB 314 para su posterior presentación en un dispositivo de visualización, tal como el dispositivo de visualización 118 de la Figura 1.

De esta manera, el decodificador de video 300 representa un ejemplo de un dispositivo de decodificación de video que incluye una o más memorias configuradas para almacenar datos de video, y uno o más procesadores configurados para recibir una imagen de datos de video, reconstruir la imagen de datos de video y aplicar un filtro basado en red neuronal (NN) a la imagen reconstruida de datos de video, en donde el filtro basado en NN comprende un filtro unificado que comprende un bloque de cabecera, un bloque de transición, uno o más bloques troncales y un bloque de cola, y en donde al menos uno del bloque de cabecera, el bloque de transición, el uno o más bloques troncales, o el bloque de cola comprende una descomposición poliádica canónica (CP) con convolución separable.

La Figura 25 es un diagrama de flujo que ilustra un ejemplo de método para codificar un bloque actual de acuerdo con las técnicas de esta divulgación. El bloque actual puede ser o incluir una CU actual. Aunque se describe con respecto al codificador de video 200 (Figuras 1 y 23),

debe entenderse que otros dispositivos pueden configurarse para realizar un método similar al de la Figura 25.

En este ejemplo, el codificador de video 200 predice inicialmente el bloque actual (400). Por ejemplo, el
5 codificador de video 200 puede formar un bloque de predicción para el bloque actual. El codificador de video 200 puede entonces calcular un bloque residual para el bloque actual (402). Para calcular el bloque residual, el codificador de video 200 puede calcular una diferencia entre el bloque
10 original, no codificado y el bloque de predicción para el bloque actual. El codificador de video 200 puede entonces transformar el bloque residual y cuantificar los coeficientes de transformación del bloque residual (404). A continuación, el codificador de video 200 puede escanear los coeficientes
15 de transformación cuantificados del bloque residual (406). Durante el escaneo, o después del escaneo, el codificador de video 200 puede codificar por entropía los coeficientes de transformación (408). Por ejemplo, el codificador de video 200 puede codificar los coeficientes de transformación usando
20 CAVLC o CABAC. El codificador de video 200 puede entonces emitir los datos codificados por entropía del bloque (410).

La Figura 26 es un diagrama de flujo que ilustra un ejemplo de método para decodificar un bloque actual de datos de video de acuerdo con las técnicas de esta
25 divulgación. El bloque actual puede ser o incluir una CU actual. Aunque se describe con respecto al decodificador de

video 300 (Figuras 1 y 24), debe entenderse que otros dispositivos pueden configurarse para realizar un método similar al de la Figura 26.

El decodificador de video 300 puede recibir datos
5 codificados por entropía para el bloque actual, tales como información de predicción codificada por entropía y datos codificados por entropía para coeficientes de transformación de un bloque residual correspondiente al bloque actual (420). El decodificador de video 300 puede
10 decodificar por entropía los datos codificados por entropía para determinar la información de predicción para el bloque actual y para reproducir coeficientes de transformación del bloque residual (422). El decodificador de video 300 puede predecir el bloque actual (424), p. ej., usando un modo de
15 intra o interpredicción como es indicado por la información de predicción para el bloque actual, para calcular un bloque de predicción para el bloque actual. El decodificador de video 300 puede entonces escanear de manera inversa los coeficientes de transformación reproducidos (426), para
20 crear un bloque de coeficientes de transformación cuantificados. El decodificador de video 300 puede entonces cuantificar de manera inversa los coeficientes de transformación y aplicar una transformación inversa a los coeficientes de transformación para producir un bloque
25 residual (428). El decodificador de video 300 puede

decodificar en última instancia el bloque actual al combinar el bloque de predicción y el bloque residual (430).

Las siguientes cláusulas numeradas ilustran uno o más aspectos de los dispositivos y técnicas descritas en esta divulgación.

Cláusula 1A. Un método para codificar datos de video, en donde el método comprende:

recibir una imagen de datos de video; reconstruir la imagen de los datos de video; y realizar un proceso de filtro basado en red neuronal (NN) sobre la imagen reconstruida de datos de video, en donde el proceso de filtro basado en NN comprende realizar una descomposición poliádica canónica (CP) con convolución separable.

Cláusula 2A. El método de la cláusula 1A, que comprende además abstenerse de realizar una extracción de características multiescala en un bloque de residuos de la imagen de datos de video.

Cláusula 3A. El método de la cláusula 1A o la cláusula 2A, eliminando además una o más capas de activación en un bloque de cabecera.

Cláusula 4A. El método de cualquiera de las cláusulas 1A-3A, en donde un grupo de entradas no comprende una activación PReLU.

Cláusula 5A. El método de la cláusula 4A, en donde el grupo de entradas comprende ReLU pero no comprende la acción PReLU.

Cláusula 6A. El método de cualquiera de las cláusulas 1A-4A, en donde el grupo de entradas no comprende ninguna activación para bloques de cabecera.

Cláusula 7A. El método de cualquiera de las
5 cláusulas 1A-6A, que comprende además abstenerse de aplicar una rama multiescala a los bloques de residuos troncales.

Cláusula 8A. El método de la cláusula 7A, que comprende además aplicar la rama multiescala a al menos un bloque de cabecera.

10 Cláusula 9A. El método de cualquiera de las cláusulas 1A-8A, en donde la descomposición CP comprende: al menos una convolución 1x1; y una circunvolución separable 3x1 en una dimensión horizontal y una dimensión vertical.

Cláusula 10A. El método de la cláusula 9A, en
15 donde la al menos una convolución 1x1 comprende: una primera convolución 1x1 que ocurre antes de las convoluciones 3x1 separables; y una segunda convolución 1x1 que ocurre después de las convoluciones separables 3x1.

Cláusula 11A. El método de la cláusula 10A, en
20 donde la segunda convolución 1x1 combina características extraídas de capas anteriores.

Cláusula 12A. El método de cualquiera de las cláusulas 1A-11A, en donde realizar la descomposición CP con convolución separable comprende realizar la
25 descomposición CP a bloques troncales y de cola.

Cláusula 13A. El método de cualquiera de las cláusulas 1A-12A, en donde realizar la descomposición CP con convolución separable comprende realizar la descomposición CP a todos los bloques.

5 Cláusula 14A. El método de cualquiera de las cláusulas 1A-13A, en donde la codificación comprende decodificación.

 Cláusula 15A. El método de cualquiera de las cláusulas 1A-14A, en donde la codificación comprende
10 codificación.

 Cláusula 16A. Un dispositivo para codificar datos de video, cuyo dispositivo comprende uno o más medios para realizar el método de cualquiera de las cláusulas 1A-15A.

15 Cláusula 17A. El dispositivo de la cláusula 16A, en donde uno o más medios comprenden uno o más procesadores implementados en circuitos.

 Cláusula 18A. El dispositivo de cualquiera de las cláusulas 16A y 17A, que comprende además una memoria
20 para almacenar los datos de video.

 Cláusula 19A. El dispositivo de cualquiera de las cláusulas 16A-18A, que comprende además una pantalla configurada para mostrar datos de video decodificados.

 Cláusula 20A. El dispositivo de cualquiera de
25 las cláusulas 16A-19A, en donde el dispositivo comprende una

o más de una cámara, un computador, un dispositivo móvil, un dispositivo receptor de difusión o un decodificador.

Cláusula 21A. El dispositivo de cualquiera de las cláusulas 16A-20A, en donde el dispositivo comprende
5 un decodificador de video.

Cláusula 22A. El dispositivo de cualquiera de las cláusulas 16A-21A, en donde el dispositivo comprende un codificador de video.

Cláusula 23A. Un medio de almacenamiento legible
10 por computador que tiene almacenadas en este instrucciones que, cuando son ejecutadas, hacen que uno o más procesadores realicen el método de cualquiera de las cláusulas 1A-15A.

Cláusula 1B. Un método de codificación o decodificación de datos de video, cuyo método comprende:
15 recibir datos para reconstruir una imagen de los datos de video; reconstruir la imagen de los datos de video; y aplicar un filtro basado en red neuronal (NN) a la imagen reconstruida de los datos de video, en donde el filtro basado en NN comprende un filtro unificado que comprende un bloque de
20 cabecera, un bloque de transición, uno o más bloques troncales y un bloque de cola, y en donde al menos uno del bloque de cabecera, el bloque de transición, el uno o más bloques troncales, o el bloque de cola comprende una descomposición poliádica canónica (CP) con convolución
25 separable.

Cláusula 2B. El método de la cláusula 1B, en donde la descomposición CP comprende: al menos una convolución 1x1; una convolución en una dimensión horizontal; y una convolución en una dimensión vertical.

5 Cláusula 3B. El método de la cláusula 2B, en donde la convolución en la dimensión horizontal comprende una de una convolución 1x3 o una convolución 3x1, y en donde la convolución en la dimensión vertical comprende otra de la convolución 1x3 o la convolución 3x1.

10 Cláusula 4B. El método de la cláusula 3B, en donde la al menos una convolución 1x1 comprende: una primera convolución 1x1 que ocurre antes de la convolución en la dimensión horizontal y la convolución en la dimensión vertical; y una segunda convolución 1x1 que ocurre después
15 de la convolución en la dimensión horizontal y la convolución en la dimensión vertical.

Cláusula 5B. El método de la cláusula 4B, en donde la segunda convolución 1x1 combina características extraídas de capas anteriores.

20 Cláusula 6B. El método de cualquiera de las cláusulas 1B-5B, en donde el uno o más bloques troncales y el bloque de cola comprenden cada uno una descomposición CP correspondiente con convolución separable.

Cláusula 7B. El método de cualquiera de las
25 cláusulas 1B-6B, en donde el bloque de cabecera, el bloque de transición, el uno o más bloques troncales y el bloque de

cola comprenden cada uno una descomposición CP correspondiente con convolución separable.

Cláusula 8B. El método de cualquiera de las cláusulas 1B-7B, en donde uno o más bloques troncales no
5 incluyen una rama multiescala.

Cláusula 9B. El método de cualquiera de las cláusulas 1B-8B, en donde el bloque de cabecera no incluye una correspondiente activación de unidad lineal rectificada paramétrica (PReLU) para al menos una entrada
10 al bloque de cabecera.

Cláusula 10B. El método de la cláusula 9B, en donde el bloque de cabecera no incluye una capa de activación correspondiente para al menos una entrada al bloque de cabecera.

15 Cláusula 11B. El método de la cláusula 10B, en donde el bloque de cabecera no incluye ninguna capa de activación.

Cláusula 12B. El método de cualquiera de las cláusulas 1B-10B, en donde el bloque de cabecera incluye al
20 menos una activación de unidad lineal rectificada (ReLU) correspondiente para al menos una entrada al bloque de cabecera.

Cláusula 13B. El método de cualquiera de las cláusulas 1B-12B, en donde el bloque de cabecera comprende
25 al menos una rama multiescala.

Cláusula 14B. El método de cualquiera de las cláusulas 1B-13B, en donde uno o más bloques troncales comprenden cada uno 24 canales.

Cláusula 15B. Un dispositivo para al menos uno
5 de codificar o decodificar datos de video, el dispositivo comprende: una o más memorias configuradas para almacenar los datos de video; y uno o más procesadores, uno o más procesadores configurados para: recibir datos para reconstruir una imagen de los datos de video; reconstruir
10 la imagen de los datos de video; y aplicar un filtro basado en red neuronal (NN) a la imagen reconstruida de datos de video, en donde el filtro basado en NN comprende un filtro unificado que comprende un bloque de cabecera, un bloque de transición, uno o más bloques troncales y un bloque de
15 cola, y en donde al menos uno del bloque de cabecera, el bloque de transición, el uno o más bloques troncales, o el bloque de cola comprende una descomposición poliádica canónica (CP) con convolución separable.

Cláusula 16B. El dispositivo de la cláusula
20 15B, en donde la descomposición CP comprende: al menos una convolución 1x1; una convolución en una dimensión horizontal; y una convolución en una dimensión vertical.

Cláusula 17B. El dispositivo de la cláusula
16B, en donde la convolución en la dimensión horizontal
25 comprende una de una convolución 1x3 o una convolución

3x1, y en donde la convolución en la dimensión vertical comprende otra de la convolución 1x3 o la convolución 3x1.

Cláusula 18B. El dispositivo de la cláusula 17B, en donde la al menos una convolución 1x1 comprende:
5 una primera convolución 1x1 que ocurre antes de la convolución en la dimensión horizontal y la convolución en la dimensión vertical; y una segunda convolución 1x1 que ocurre después de la convolución en la dimensión horizontal y la convolución en la dimensión vertical.

10 Cláusula 19B. El dispositivo de la cláusula 18B, en donde la segunda convolución 1x1 combina características extraídas de capas anteriores.

Cláusula 20B. El dispositivo de cualquiera de las cláusulas 15B-19B, en donde uno o más bloques troncales
15 y el bloque de cola comprenden cada uno una descomposición CP correspondiente con convolución separable.

Cláusula 21B. El dispositivo de cualquiera de las cláusulas 15B-20B, en donde el bloque de cabecera, el bloque de transición, el uno o más bloques troncales, y el bloque
20 de cola comprenden cada uno una descomposición CP correspondiente con convolución separable.

Cláusula 22B. El dispositivo de cualquiera de las cláusulas 15B-21B, en donde uno o más bloques troncales no incluyen una rama multiescala.

25 Cláusula 23B. El dispositivo de cualquiera de las cláusulas 15B-22B, en donde el bloque de cabecera no

incluye una correspondiente activación de unidad lineal rectificada paramétrica (PReLU) para al menos una entrada al bloque de cabecera.

Cláusula 24B. El dispositivo de la cláusula 23B,
5 en donde el bloque de cabecera no incluye una capa de activación correspondiente para al menos una entrada al bloque de cabecera.

Cláusula 25B. El dispositivo de la cláusula 24B, en donde el bloque de cabecera no incluye ninguna
10 capa de activación.

Cláusula 26B. El dispositivo de cualquiera de las cláusulas 15B-24B, en donde el bloque de cabecera incluye al menos una activación de unidad lineal rectificada (ReLU) correspondiente para al menos una
15 entrada al bloque de cabecera.

Cláusula 27B. El dispositivo de cualquiera de las cláusulas 15B-26B, en donde el bloque de cabecera comprende al menos una rama multiescala.

Cláusula 28B. El dispositivo de cualquiera de las
20 cláusulas 15B-27B, en donde uno o más bloques troncales comprenden cada uno 24 canales.

Cláusula 29B. El dispositivo de cualquiera de las cláusulas 15B-28B, que comprende además una pantalla configurada para mostrar datos de video decodificados.

25 Cláusula 30B. El dispositivo de cualquiera de las cláusulas 15B-29B, en donde el dispositivo comprende una

o más de una cámara, un computador, un dispositivo móvil, un dispositivo receptor de difusión o un decodificador.

Cláusula 31B. Un dispositivo para al menos uno de codificación o decodificación de datos de video, cuyo
5 dispositivo comprende: medios para recibir datos para reconstruir una imagen de los datos de video; medios para reconstruir la imagen de los datos de video; y medios para aplicar un filtro basado en red neuronal (NN) a la imagen reconstruida de datos de video, en donde el filtro basado en
10 NN comprende un filtro unificado que comprende un bloque de cabecera, un bloque de transición, uno o más bloques troncales y un bloque de cola, y en donde al menos uno del bloque de cabecera, el bloque de transición, el uno o más bloques troncales, o el bloque de cola comprende una
15 descomposición poliádica canónica (CP) con convolución separable.

Cláusula 32B. Medios de almacenamiento no transitorios que comprenden instrucciones que, cuando se ejecutan, hacen que uno o más procesadores: reciban datos
20 para reconstruir una imagen de datos de video; reconstruir la imagen de los datos de video; y aplicar un filtro basado en red neuronal (NN) a la imagen reconstruida de datos de video, en donde el filtro basado en NN comprende un filtro unificado que comprende un bloque de cabecera, un bloque
25 de transición, uno o más bloques troncales y un bloque de cola, y en donde al menos uno del bloque de cabecera, el

bloque de transición, el uno o más bloques troncales, o el bloque de cola comprende una descomposición poliádica canónica (CP) con convolución separable.

Se debe reconocer que dependiendo del ejemplo, ciertos actos o eventos de cualquiera de las técnicas divulgadas en la presente pueden realizarse en una secuencia diferente, pueden añadirse, fusionarse o dejarse fuera por completo (p. ej., no todos los actos o eventos descritos son necesarios para la práctica de las técnicas). Además, en ciertos ejemplos, los actos o eventos pueden realizarse simultáneamente, p. ej., a través de procesamiento multiproceso, procesamiento de interrupción o múltiples procesadores, en lugar de secuencialmente.

En uno o más ejemplos, las funciones descritas pueden implementarse en hardware, software, firmware o cualquier combinación de estos. Si se implementan en software, las funciones pueden almacenarse o transmitirse como una o más instrucciones o código en un medio legible por computador y ejecutarse mediante una unidad de procesamiento basada en hardware. Los medios legibles por computador pueden incluir medios de almacenamiento legibles por computador, que corresponden a un medio tangible tal como medios de almacenamiento de datos, o medios de comunicación que incluyen cualquier medio que facilite la transferencia de un programa informático de un lugar a otro, p. ej., de acuerdo con un protocolo de comunicación. De

esta manera, los medios legibles por computador generalmente pueden corresponder a (1) medios de almacenamiento legibles por computador tangibles que no son transitorios o (2) un medio de comunicación tal como una
5 señal u onda portadora. Los medios de almacenamiento de datos pueden ser cualquier medio disponible al que se pueda acceder por uno o más computadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para la implementación de las técnicas divulgadas en
10 esta descripción. Un producto de programa informático puede incluir un medio legible por computador.

A manera de ejemplo, y no de limitación, tales medios de almacenamiento legibles por computador pueden incluir uno o más de RAM, ROM, EEPROM, CD-ROM u otro
15 almacenamiento en disco óptico, almacenamiento en disco magnético u otros dispositivos de almacenamiento magnético, memoria flash, o cualquier otro medio que pueda usarse para almacenar el código de programa deseado en forma de instrucciones o estructuras de datos y al que se pueda
20 acceder por un computador. Además, cualquier conexión se denomina correctamente un medio legible por computador. Por ejemplo, si las instrucciones se transmiten desde un sitio web, servidor u otra fuente remota usando un cable coaxial, cable de fibra óptica, par trenzado, línea de suscriptor
25 digital (DSL) o tecnologías inalámbricas tal como infrarroja, radio y microondas, entonces el cable coaxial,

cable de fibra óptica, par trenzado, DSL, o tecnologías inalámbricas tal como infrarroja, radio y microondas se incluyen en la definición de medio. Sin embargo, debe entenderse que los medios de almacenamiento legibles por computador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros medios transitorios, sino que se dirigen a medios de almacenamiento tangibles no transitorios. Disco, como se usa en la presente, incluye disco compacto (CD, por sus siglas en inglés), disco láser, disco óptico, disco versátil digital (DVD, por sus siglas en inglés), disquete y disco Blu-ray, donde los discos generalmente reproducen datos magnéticamente, mientras que los discos reproducen datos ópticamente con láseres. Las combinaciones de lo anterior también deben incluirse dentro del alcance de los medios legibles por computador.

Las instrucciones pueden ejecutarse por uno o más procesadores, tales como uno o más DSP, microprocesadores de propósito general, ASIC, FPGA u otros circuitos lógicos integrados o discretos equivalentes. En consecuencia, los términos “procesador” y “circuito de procesamiento”, tal como se usan en la presente, pueden referirse a cualquiera de las estructuras anteriores o cualquier otra estructura adecuada para la implementación de las técnicas descritas en la presente. Además, en algunos aspectos, la funcionalidad descrita en la presente puede proporcionarse dentro de

módulos de hardware y/o software dedicados configurados para codificación y decodificación, o incorporarse en un códec combinado. También, las técnicas podrían implementarse completamente en uno o más circuitos o elementos lógicos.

5 Las técnicas de esta divulgación pueden implementarse en una amplia variedad de dispositivos o aparatos, incluidos un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de CI (p. ej., un conjunto de chips). En esta divulgación se describen diversos
10 componentes, módulos o unidades para enfatizar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no necesariamente requieren realización por diferentes unidades de hardware. Más bien, como se describió anteriormente, varias unidades pueden
15 combinarse en una unidad de hardware de códec o proporcionarse mediante una colección de unidades de hardware interoperativas, incluidos uno o más procesadores como se describió anteriormente, junto con software y/o firmware adecuados.

20 Se han descrito diversos ejemplos. Estos y otros ejemplos están dentro del alcance de las siguientes reivindicaciones.