


Article

An Image-Retrieval Method Based on Cross-Hardware Platform Features

Jun Yin ¹, Fei Wu ¹ and Hao Su ^{2,*} ¹ School of Computer Science and Technology, Zhejiang University, Hangzhou 310058, China; 12221242@zju.edu.cn (J.Y.); wufei@zju.edu.cn (F.W.)² Zhejiang Dahua Technology Co., Ltd., Hangzhou 310053, China

* Correspondence: chaozhong2010@163.com

Abstract: Artificial intelligence (AI) models have already achieved great success in fields such as computer vision and natural language processing. However, deploying AI models based on heterogeneous hardware is difficult to ensure accuracy consistency, especially for precision sensitive feature-based image retrieval. In this article, we realize an image-retrieval method based on cross-hardware platform features, aiming to prove that the features of heterogeneous hardware platforms can be mixed, in which the Huawei Atlas 300V and NVIDIA TeslaT4 are used for experiments. First, we compared the decoding differences of heterogeneous hardware, and used CPU software decoding to help hardware decoding improve the decoding success rate. Then, we compared the difference between the Atlas 300V and TeslaT4 chip architectures and tested the differences between the two platform features by calculating feature similarity. In addition, the scaling mode in the pre-processing process was also compared to further analyze the factors affecting feature consistency. Next, the consistency of capture and correlation based on video structure were verified. Finally, the experimental results reveal that the feature results from the TeslaT4 and Atlas 300V can be mixed for image retrieval based on cross-hardware platform features. Consequently, cross-platform image retrieval with low error is realized. Specifically, compared with the Atlas 300V hard and CPU soft decoding, the TeslaT4 hard decoded more than 99% of the image with a decoding pixel maximum difference of +1/−1. From the average of feature similarity, the feature similarity between the Atlas 300V and TeslaT4 exceeds 99%. The difference between the TeslaT4 and Atlas 300V in recall and mAP in feature retrieval is less than 0.1%.



Citation: Yin, J.; Wu, F.; Su, H. An Image-Retrieval Method Based on Cross-Hardware Platform Features. *Appl. Syst. Innov.* **2024**, *7*, 64. <https://doi.org/10.3390/asi7040064>

Academic Editors: Patrícia Ramos and Jose Manuel Oliveira

Received: 11 June 2024

Revised: 18 July 2024

Accepted: 19 July 2024

Published: 23 July 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: heterogeneous hardware; Huawei Atlas 300V; NVIDIA TeslaT4; image retrieval; cross-hardware

1. Introduction

Nowadays, artificial intelligence (AI) models have been widely used in autonomous driving, medical image analysis, robotics, smart security, and natural language processing, etc. These AI models not only have a huge number of parameters, but the data required for training are also extremely large, thus placing unprecedented demands on computing power, such as transformer and convolutional neural networks (CNN). To overcome this problem, many companies and researchers use model compression, distributed computing, heterogeneous computing, and other technologies to alleviate the pressure of single-point computing [1–3]. Other companies and researchers design a number of acceleration chips to improve the performance of training or inferencing in the above areas [4–6].

AI chip architecture is designed to efficiently implement machine-learning (ML) algorithms, specifically for parallel computing of deep neural networks (DNNs). The current mainstream architectures include GPU, FPGA, and ASIC. GPU architecture is widely used to train large deep-learning models because of its highly parallel computing capability. In the GPU field, NVIDIA GPUs are widely used in deep-learning training and inference tasks due to their outstanding performance and versatility, which promotes the rapid

development of the deep-learning field. Also, NVIDIA GPU-based AI models have been widely deployed in data centers and edge devices, such as the A100, H100, Tesla T4, and Jetson [7–10]. The field programmable gate array (FPGA) adapts to a variety of AI algorithms through flexible programming, and has a high energy efficiency ratio, which has advantages in specific scenarios [11,12].

The application-specific integrated circuit (ASIC) is a custom integrated circuit designed for a specific application, and it often has significant advantages over general-purpose chips (such as GPUs, CPUs) in terms of task efficiency, power consumption, size, and cost. Google's self-developed tensor processing unit (TPU), an ASIC specifically optimized for machine learning and artificial intelligence, has played an important role in Google's internal TensorFlow framework, significantly increasing the speed of deep-learning training and inferencing [13]. The Apple Neural Engine (ANE) is an efficient, high-throughput inference engine for machine learning. It will minimize the impact of machine-learning inference workloads on application responsiveness, application memory, and device battery life [14]. With the brand-new Qualcomm[®] Hexagon[™] Tensor Accelerator, the Qualcomm AI Engine pushes 15 trillion operations per second with maximum efficiency [15]. Focusing on AI system-on-chip (SoC) dedicated processors, Kneron aims to provide end-users with high-performance, low-power, low-cost solutions through edge AI based on a self-developed neural processing unit (NPU) accelerated neural network model that can be applied to AI applications in various end-devices [16]. The Huawei Ascend series chips are high energy efficiency, flexible, and programmable artificial intelligence processors, using the self-developed Huawei DaVinci architecture, integrating rich computing units to improve AI computing integrity and efficiency and expand the applicability of the chip [17,18].

In actual inference deployment, due to factors such as chip cost and power consumption, the current mainstream AI model deployment scheme usually adopts heterogeneous hardware for AI inference. Some researchers compared the more mainstream deep-learning computing cards currently on the market. Deng et al. [19] studied the more mainstream deep-learning computing cards in the current market, from the four aspects of computing power, memory performance, power-consumption efficiency, and parallel computing, providing a theoretical basis for the selection of computing cards. Lu et al. [20] systematically evaluated and analyzed Ascend chips, including the performance and power comparison between Ascend and GPU, Ascend deep-learning framework, operators, and mixed-precision training-optimization strategies. Kum et al. [21] improved the utilization of GPU resources by adjusting the batch size of inference application input, so as to realize real-time video analysis based on deep learning. Goel et al. [22] used hierarchical DNNs to create a parallel inference pipeline for higher throughput in computer vision problems.

With the continuous development of deep-learning technology, image-retrieval technology based on feature extraction has made a lot of progress and has gradually become an indispensable part of structured video data processing. Image-retrieval technology can quickly search high-value objects such as non-motor vehicles and vehicles in image datasets and has broad application prospects and practical value in traffic-situation analysis, crowd-density estimation, and cluster and image retrieval. Video structuring is a technology for extracting video content information. It organizes video content into structured information that can be understood by computers and people and semi-structured information that can be compared according to semantic relationship by means of spatiotemporal segmentation, feature extraction, object recognition. Video structuring can capture and extract structured information of motor vehicles, license plates, non-motor vehicles, and other information in video-streaming media in real time, aiming at efficient extraction and structured storage of massive video information within a certain spatiotemporal range, greatly reducing the spatiotemporal range of retrieval, achieving structured data research and judgment of specific targets or large amounts of data, and improving storage and retrieval efficiency. It has high practical value for target trajectory research, target quick search, traffic-situation analysis, privacy protection, and other directions.

Feature extraction in image-retrieval technology requires a lot of computing resources, so engineers use multi-chip hybrid deployment to extract features. However, due to differences in AI inference chips, it is difficult to ensure consistency of accuracy when deploying AI models based on heterogeneous hardware, especially for feature-based image retrieval, which is sensitive to accuracy [23–26]. Therefore, it is particularly important to verify the accuracy consistency of heterogeneous hardware and reduce the loss of accuracy across platforms. After verifying the feature consistency of heterogeneous hardware, the features of different hardware platforms can be directly used in production without considering the differences between hardware, which further expands the deployment range of algorithm schemes and has significant practical value.

In feature extraction, the main factors that affect feature quality include codec, color-space conversion, image scaling, model inference (target bounding box, model quantization, algorithm scheme model version, image matting, operator fusion), etc. In this paper, the AI model uses FP16 quantization, which can strike a good balance between precision and speed [8,17].

In this paper, we focus on the implementation of an image-retrieval method based on cross-hardware platform features, which reduces the cross-platform accuracy loss, and proves that the features of heterogeneous hardware platforms can be mixed, in which the Huawei Atlas 300V and NVIDIA TeslaT4 are used for experiments. First, we compare the decoding differences of heterogeneous hardware, and use CPU software decoding to help hardware decoding improve the decoding success rate. Also, the color-space conversion module is added to further improve the image utilization rate. Then, we compared the difference between the Atlas 300V and TeslaT4 chip architectures and tested the differences between the two platform features by calculating feature similarity. In addition, the scaling mode in the pre-processing process is also compared to further analyze the factors affecting the feature consistency. Next, the consistency of capture and correlation based on video structure is verified. Finally, the experimental results reveal that the feature results from the TeslaT4 and Atlas 300V can be mixed for image retrieval based on cross-hardware platform features. Consequently, cross-platform image retrieval with low error is realized.

2. Related Work

Content-based image retrieval (CBIR) has long been an important research topic in the field of computer vision [27–31]. Image-retrieval technology based on image content semantics uses the color and texture of the image as well as the information of the image object and category to retrieve the image similar to it in the image-retrieval database. Prerna et al. [26] proposed a local pentagon pattern integrating heterogeneous features, which can accurately extract heterogeneous patterns of images, thus improving the accuracy of image retrieval and image recognition. Muthusami et al. [32] came up with a semantic-based clustering method to support a visual query of heterogeneous features in images. In addition, the heterogeneous manifold ranking (HMR) method proposed by Wu et al. [33] aims to enhance image-retrieval performance by leveraging the complementary information between image-related click data and their visual content, thereby effectively addressing the inherent challenges of noise and sparsity in click data. The improved method proposed by Wang et al. [34] is based on twin features (TF) and maximum similarity matching (SMM), aiming to enhance the quality of feature matching. Additionally, they have designed and implemented an efficient image-retrieval system utilizing a cloud heterogeneous computing framework. Yu et al. [35] propose a heterogeneous image-retrieval system, which combines a high-precision digital system with a highly parallel memory search to improve the efficiency of image retrieval.

In recent years, feature-extraction methods based on deep learning have become dominant in the field of image retrieval. In particular, convolutional neural networks (CNNs) have played an important role in significantly improving the accuracy of image retrieval based on features by representing images as compact vectors [36–39]. Zhang et al. [40] proposed a new fabric image-retrieval methodology based on the fusion of multi-modal

feature, and established a benchmark fabric-image database for multi-modal retrieval. Zhan et al. [41] proposed a “coarse-to-fine” image-retrieval strategy, which uses the fusion of global features extracted by CNNs and local features extracted by SIFT to match the most relevant images. Ye et al. [42] proposed a remote-sensing image-retrieval method based on weighted distance and a convolutional neural network (CNN) to extract features, which significantly improved the accuracy of remote-sensing image retrieval. To address the risks of CNNs’ feature-based image retrieval, Fan et al. [43] designed a computable encryption scheme based on vector and matrix calculations. The vision transformer developed in recent years has also achieved outstanding results in image retrieval. Lv et al. [44] used knowledge distillation to compress the Transformer framework to reduce computational complexity for efficient image retrieval. Li et al. [45] utilized Transformer encoders to handle the complex dependencies between visual features and labels, enabling multi-label image retrieval. In order to realize the fusion of global features and local features, Li et al. [46] proposed a multi-scale feature-fusion image-retrieval method (MSViT) based on visual transformation.

In recent years, many researchers have deployed artificial intelligence (AI) models on accelerators that are ASIC- or FPGA-based [47,48]. Kalapothis et al. [49] implemented dynamic deployment of AI models on the Kria KV260 development platform, thus successfully using hardware-accelerated unit inference AI models. Nechi et al. [50] summarized the current state of deep-learning hardware acceleration and evaluated more than 120 FPGA-based neural-network accelerator designs. Wu et al. [51] studied the neural networks involved in FPGA-based acceleration systems. Machupalli et al. [52] provided a review of existing deep-neural-network ASIC accelerators and classified them according to the optimization techniques used in their implementation. Tang et al. [53] studied the application performance of ASIC CXL in various data center scenarios and provided a benchmark.

3. Methods

The overall process of feature-based image retrieval is shown in Figure 1, and includes the decoding module, detection module, and feature extraction module, in which the video stream includes the target tracking, target selection, and capture modules. Firstly, we verified the metrics differences between the two platforms in image retrieval based on features. Then, we designed a decoding comparison experiment to verify the decoding difference between the Atlas 300V and TeslaT4 and used CPU software for decoding to improve the decoding success rate of both. Finally, we compared the difference between the Atlas 300V and TeslaT4 chip architectures, and used 1v1 to test the consistency of the two platform models.

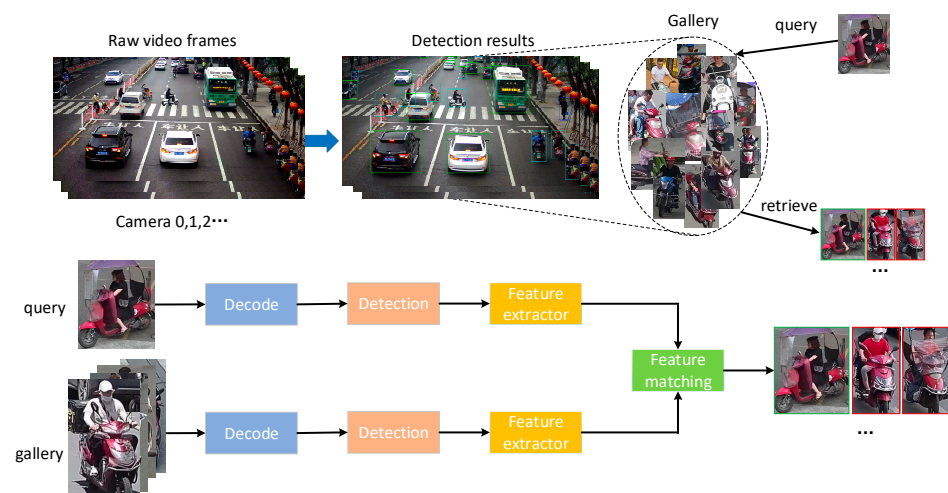


Figure 1. The overall process of feature-based image retrieval. The green bounding box indicates the correct target, and the red bounding box indicates false alarms.

3.1. Image Retrieval Based on Cross-Hardware Platform Features

The quality of the feature will affect the image-retrieval results based on features. Data-structuring technology is mainly used to extract structural features and a build feature database. Due to the differences in hardware platforms, we have implemented different deployment schemes for data structuring. The deployment schemes based on the Tesla T4 and Atlas 300V is shown in Figure 2. For video streaming, the deployment scheme based on the Tesla T4 first uses a hardware acceleration unit to decode the streaming media, then down-samples the video frame, then performs target detection and tracking, selects the best quality targets to capture and perform attribute analysis and feature extraction, and finally encodes the captured images and stores the analysis results and captured images into the database. However, due to memory limitations and optimal performance, the deployment schemes based on the Atlas 300V first need to send the streaming media from the host to the device through PCIe, then use the hardware acceleration unit to decode the streaming media, then down-sample the video frame, then perform target detection and tracking, then transmit the video frame encoding to the host, and select the best quality target to capture. Then, the captured image is transmitted to the device for decoding, attribute analysis, and feature extraction, and finally the analysis results and captured images are stored in the database. For images, the deployment schemes based on the Tesla T4 first decode them using a hardware acceleration unit, then perform object detection, attribute analysis, and feature extraction, and finally stores the results into a database. In addition, the Atlas 300V-based deployment scheme first sends the image from the host to the device via PCIe, then decodes the image using a hardware acceleration unit, then performs object detection, attribute analysis, and feature extraction, and finally sends the results back from the device to the host and stores them in the database.

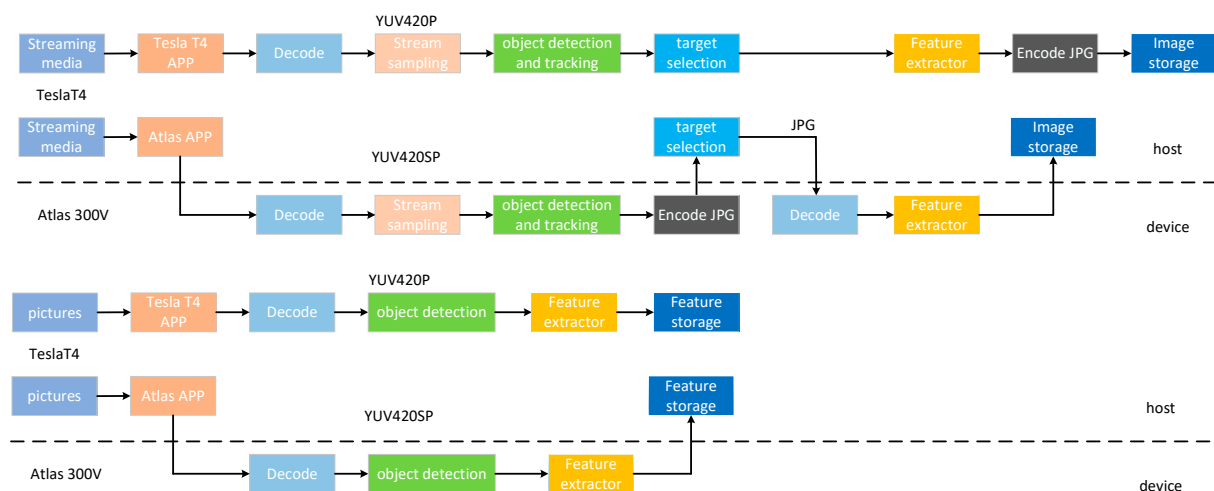


Figure 2. The deployment schemes based on the Tesla T4 and Atlas 300V.

Host refers to the X86 server and ARM server connected to the device, which uses the neural network (NN) computing capability provided by the device to complete services. A device is a hardware device installed with an Ascend AI processor and connected to a host through a PCIe port to provide NN computing capabilities [54,55].

The YUV format has a small amount of data and strong anti-noise, which is conducive to image compression and transmission and reduces bandwidth consumption [56]. According to the chip whitepaper, the Tesla T4 is in YUV420P format [7,8] and the Atlas 300V is in YUV420SP format [17,54]. Since the image is encoded by the Joint Photographic Experts Group (JPEG), in order to reduce the feature difference caused by image quality loss, we set the image coding quality factor to 95.

Video structuring is a technology for extracting video content information. It organizes video content into structured information that can be understood by computers and people

and semi-structured information that can be compared according to semantic relationships by means of spatiotemporal segmentation, feature extraction, and object recognition. Video structuring can capture and extract structured information of motor vehicles, license plates, non-motor vehicles, and other information in video-streaming media in real time, aiming at efficient extraction and structured storage of massive video information within a certain spatiotemporal range, greatly reducing the spatiotemporal range of retrieval, achieving structured data research and judgment of specific targets or large amounts of data, and improving storage and retrieval efficiency. It has high practical value for target trajectory research, target quick search, traffic-situation analysis, privacy protection and other directions.

With the continuous development of deep-learning technology, image-retrieval technology based on feature extraction has made a lot of progress and has gradually become an indispensable part of structured video data processing. Image-retrieval technology can quickly search high-value objects such as non-motor vehicles and vehicles in image datasets and has broad application prospects and practical value in traffic-situation analysis, crowd-density estimation, and cluster and image retrieval.

Since the results of each module are not 100% consistent, whether because of a noise amplification effect (large deviation of feature search results) or noise cancellation (consistent feature search results) in the feature-search system, we constructed a feature-search experiment. First of all, we use video-structuring technology to score the quality of objects, such as riders, motor vehicles, and non-motor vehicles, in video images, capture the optimal target, and extract the features from the target capture results to form an image database. Each image is mapped with feature data. Then, a large number of interference images are detected, their features are extracted, and the feature results are randomly mixed into the feature-retrieval database to test the robustness of the system. Next, the image to be searched is detected and the features of the key target are extracted. The cosine similarity is calculated one by one with the images captured in the feature database, and the calculated similarity results are ranked from highest to lowest. The higher the ranking, the higher the similarity. Thus, the function of retrieving the same target image from different points in the feature-based database is realized. Finally, we extract features based on the TeslaT4 and Atlas 300V, respectively, to establish feature databases, and input images to be retrieved for feature retrieval to verify the differences between the two platforms. Also, we set the similarity threshold to 0.3, and calculated recall and mAP under Top 200 as evaluation results.

3.2. Decoding Analysis

The decoding difference will significantly affect the quality of feature extraction. The quantization matrix, HUFFMAN table, and DCT/IDCT process used by different JPEG codecs are not exactly the same, and the output codec results are often different. This paper mainly compares the decoding hardware capabilities.

The Turing-based NVIDIA Tesla T4 supports HEVC 4:4:4 (8/10/12 bit), H.264, HEVC 4:2:0 (8/10/12 bit), VP9 (8/10/12 bit) and other video decoding formats. In addition, the JPEGD decoding of the TeslaT4 chip supports the maximum resolution of the input image, $32,768 \times 16,384$, and the minimum resolution, 64×64 [7,8].

For Atlas 300V, the JPEGD input image maximum resolution is 8192×8192 , and the minimum resolution is 32×32 . When the Atlas 300V implements the JPEGD image decoding function, it only supports Huffman encoding. The color space of the original image before compression is YUV, and the ratio of each component of the pixel is 4:4:4, 4:2:2, 4:2:0, 4:0:0, or 4:4:0. Arithmetic encoding, progressive JPEG format, and JPEG2000 format are not supported. In addition, the Atlas 300V supports a maximum of four Huffman tables on hardware, including two direct current tables and two alternating current tables. Ascend supports 3 quantization tables, 8-bit sampling accuracy, and decoding of sequential encoded images. Also, it supports JPEG format decoding based on discrete cosine transform (DCT) and supports image decoding with a start of scan (SOS) flag. Video decoder (VDEC)

only supports decoding of input streams by frame. If there are bad frames or missing frames in the code stream, the decoder VDEC will mark the frame as a decoding failure and report an exception. Through the interlacing mode encoding out of the stream, VDEC only supports decoding H264 8-bit encoded stream. At the same time, it can achieve the purpose of fast decoding and fast output, but this scenario does not support decoding the code stream containing B frames [17,54].

Due to the limitations of the hardware of various manufacturers, the pictures and videos of many scenes could be directly decoded successfully. By analyzing the image decoding failure, we found that a few of the image decoding failures were due to an image format error, and the image is opened and presented as black. Most of the image failures were caused by image resolutions, width and height alignments, image memories, etc., that do not meet the hardware platform. Other image-decoding failures are due to the limitations of the hardware itself, resulting in inconsistent images successfully decoded on both sides.

In order to reduce the difference in decoding, we first compared the differences between hardware-platform decoding and CPU software decoding. Specifically, the same images and videos were decoded through different decoding modules to obtain their respective YUV images, the absolute value of the difference between the two was calculated pixel by pixel, and then the number of current differences was calculated, respectively, according to the absolute value of the pixel difference. Next, we chose the software decoding mode with the closest precision to the hardware decoding and executed the software decoding directly after the hardware decoding failed. In addition, we transplanted the different modes of hardware decoding and software decoding (such as the 3 DCT/IDCT modes in Turbo jpeg) into the software decoding, using the same DCT mode, so as to achieve the same codec results. Finally, we added a color-space conversion module to convert input formats not supported by the hardware pre-processing module to formats it supports (such as YUV444 to YUV420), so as to further improve the image utilization.

3.3. Hardware Architecture (GPU/NPU)

The hardware architecture difference will significantly affect the quality and speed of feature extraction. In this paper, the Tesla T4 (GPU) and Atlas 300V (NPU) are selected for comparison.

As shown in Figure 3, GPU architecture has been widely used in the field of artificial intelligence because of its high parallel computing capability and versatility. In GPU fields, NVIDIA GPUs are widely used in deep-learning training and inference tasks due to their outstanding performance and versatility, which promotes the rapid development of the deep-learning field. In addition, a large number of enterprises are adopting NVIDIA GPUs to deploy AI models in data centers and edge devices. The NVIDIA Tesla T4 further extends this lead, from FP32 to FP16 to INT8 and INT4, delivering breakthrough performance and flexible multi-precision capabilities. Specifically, the NVIDIA T4 is an AI inference accelerator based on the new NVIDIA Turing™ architecture, featuring high performance, low latency, multi-precision, and hybrid accuracy. The NVIDIA Tesla T4 GPU includes 2560 CUDA cores and 320 tensor cores, which deliver FP16 performance of up to 65 TFLOPS and INT8 inference performance of up to 130 TOPS. In addition, the NVIDIA Tesla T4 introduces a revolutionary Turing tensor core technology that uses multi-precision computation for different workloads. The NVIDIA Tesla T4 accelerates different cloud workloads, including video transcoding, deep-learning training, and inference, to deliver revolutionary performance and innovative solutions [7,8,57]. Turing GPUs have a CUDA Compute Capability of 7.5.

An NPU is a kind of application-specific integrated circuit (ASIC), in which the processing of the network is more complex and more flexible, and generally can use software or hardware, according to the characteristics of network computing special programming, to achieve the special purpose of the network. NPUs in a chip can achieve many different

functions, and are applicable to a variety of different network equipment and products. Huawei Ascend AI processor is a successful example of NPU architecture design.



Figure 3. GPU architecture [57,58].

The Huawei Ascend AI processor is essentially a system-on-chip, which can mainly be applied to scenes related to voice, word, video, and image processing. Its main architectural components include special corresponding control units, large capacity storage units, and computing units [18].

As is shown in Figure 4, the Ascend core, based on the DaVinci architecture, consists of an instruction management unit, multi-level on-chip memory, and corresponding loading/storage units, three types of computing units, etc. Specifically, the computing unit mainly includes scalar, vector, and cube. The scalar unit is similar to the classical RISC core and is primarily responsible for the operations of the control flow and scalar calculations. The 1D vector unit is similar to the SIMD unit, which can perform most computation operations, such as format transfer, activation, normalize, CV operators (RPN, etc.). The 2D matrix unit is used to perform general matrix–matrix multiplication (GEMM), including dot product and outer product, because it can reuse intermediate data and accelerate computational efficiency. The 3D cube unit then provides a lot of computing power to perform Mat-Mul, FC, convolution, etc. The cube unit can alleviate the mismatch between limited memory bandwidth and computational throughput due to its excellent data reuse capability [18]. The Ascend core uses multi-level buffers to ensure data reuse. For example, L0 buffers are dedicated to 3D cube units. In addition, vector units are also responsible for data accuracy conversion, such as quantization and de-quantization operations between int32, fp16, and int8 [18].

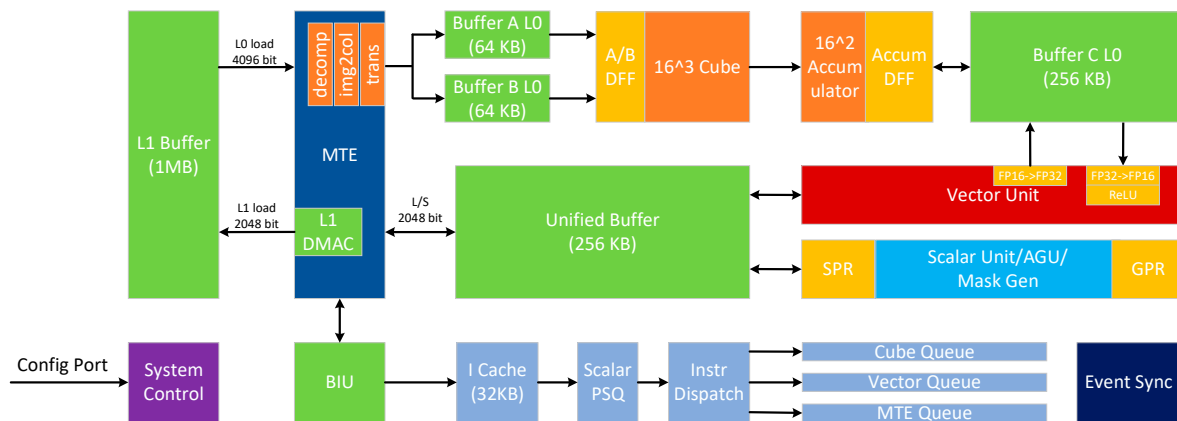


Figure 4. Ascend core (Ascend-Max configuration) [9] (Copyright © 2021, IEEE).

In addition, the compute architecture for neural networks (CANN) is a heterogeneous computing architecture launched by Huawei for AI scenarios, and its overall architecture is shown in Figure 5. The CANN provides multi-level programming interfaces and supports the rapid construction of AI applications and services based on the Ascend with the advantages of full scenario, low threshold, and high performance, which improve user-development efficiency and release the surging computing power of the Ascend AI processor. Also, the components of CANN include the Ascend computing language (AscendCL), graph engine (GE), runtime, DVPP, CANN Lib, HCCL, driver, development system, system tools, and auxiliary development tools [54,55].

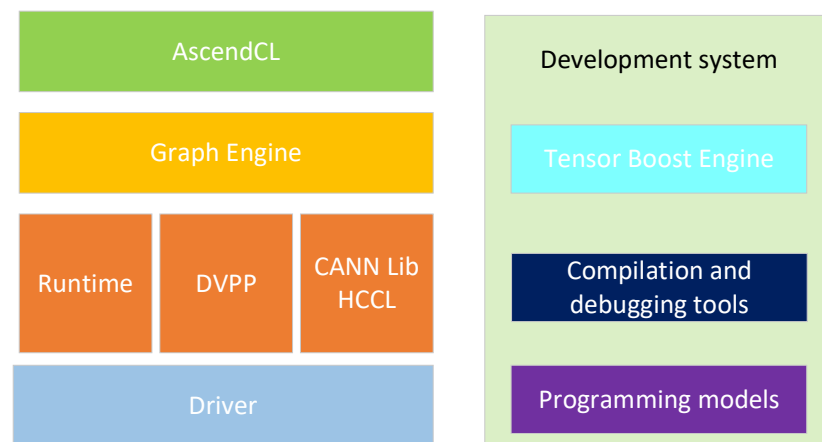


Figure 5. Illustration of Ascend's software development stack [55].

The CANN [54,55] mainly provides efficient tensor computing power, supporting a variety of data types and operations, including matrix multiplication, convolution, pooling, etc. It also supports a variety of optimization techniques, such as dynamic memory management and quantitative computation, which can greatly improve the training and reasoning speed of neural networks. In terms of hardware architecture, the CANN adopts the architecture design of high parallelism in the computing core and cache, supports a variety of data parallelism and model parallelism, and can meet the computing requirements of neural networks of different scales. At the same time, CANN also supports a variety of programming languages and development tools, such as C/C++, Python, etc., to facilitate developers to develop and deploy applications. Also, the CANN platform provides a series of tools and libraries, including model transformation tools, inference engines, optimization libraries, etc., to help developers deploy neural-network models on Huawei Ascend series chips to achieve high-performance inference computing. Through these tools and libraries,

developers can more easily apply neural networks to a variety of practical scenarios, such as natural language processing, speech recognition, image recognition, etc.

The product specifications of the Atlas 300V and Tesla T4 are compared in Table 1. As shown in Figure 6, in order to verify the difference between the Atlas 300V and Tesla T4 in the accuracy of inference models, we constructed their test schemes based on the feature-extraction processes of video stream and image stream, respectively. The detection models (such as Yolo [59–62]) of video streams and image streams are different, but the feature-extraction models (such as ResNet [63], ResNeXt [64], MobileNets [65], ShuffleNet [66,67], and Inception-v4 [68]) are the same. Therefore, we pay more attention to the result of the capture in the video stream.

Table 1. Comparison among hardware platforms [7,17,69].

Platform	TeslaT4	Atlas 300V	Atlas 300V Pro
Architecture	NVIDIA Turing	DaVinci	DaVinci
Memory size	16 GB GDDR6	24 GB LPDDR4X	48 GB LPDDR4X
Memory BW	320 GByte/s	204.8 GByte/s	204.8 GByte/s
Power	70 W	72 W	72 W
PCIe	×16 PCIe Gen3	×16 PCIe Gen4	×16 PCIe Gen4
FP16	65 TFLOPS	50 TFLOPS	70 TFLOPS
INT8	130 TOPS	100 TOPS	140 TOPS
H.264/H.265 decoder	-	100 channels 1080P 25FPS, 80 channels 1080P 30FPS, or 10 channels 4K 60FPS	128 channels 1080P 30FPS or 16 channels 4K 60FPS
H.264/H.265 encoder	-	24 channels 1080P 30FPS or 3 channels 4K 60FPS	24 channels 1080P 30FPS or 3 channels 4K 60FPS
JPEG decoder	-	4K 512FPS	4K 512FPS
JPEG encoder	-	4K 256FPS	4K 256FPS

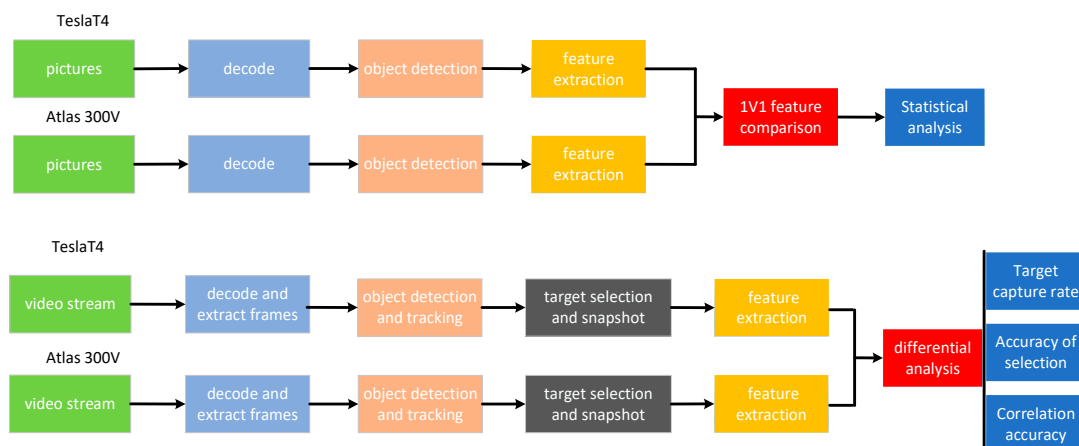


Figure 6. Feature-extraction and comparison methods in images and video streams.

First, we converted the AI model to the FP16 model using the respective platform's deployment software, which can strike a good balance between precision and speed. Then, we conducted the test according to the test scheme of the video stream in Figure 6 and calculated the test index, in which 60 videos using H264 and MPEG4 encoding were tested, with resolutions ranging from 1080p to 4K and video frame rates of 25 frames per second. We sampled the video frame rate from 25 frames per second to 8 frames per second. Also, we focused on the difference between the target capture rate and the target association rate within 1%, and our task results are consistent. Next, the model deployment for image streams is the same as for video streams. A large number of JPEG images are detected and

then feature extraction is performed on the detected targets. Then, the cosine similarity formula is used to calculate the similarity of the two feature vectors,

$$\cos(\theta) = \frac{\vec{a} \bullet \vec{b}}{\|\vec{a}\| \|\vec{b}\|} \quad (1)$$

where \vec{a} , \vec{b} represent vectors.

Finally, the statistical index of eigenvector similarity is calculated.

4. Results

In this section, we compare the differences between the TeslaT4 and Atlas 300V in three typical AI computing scenarios, including decoding, feature extraction, and feature retrieval. The TeslaT4 deployment environments are CUDA10.1, libcudnn7.6.3, and tensorRT-6.0.1.5. The Atlas 300V is deployed on CANN6.0.1. Our test data are from the actual production environment, such as park entrance, sidewalk, non-motorized lane, traffic vehicle bayonet, and other scenes.

4.1. Image-Retrieval Result Based on Cross-Hardware Platform Features

In the feature-based image retrieval, we set the feature matching threshold to 0.3 and compared the feature search results of the Atlas 300V and TeslaT4 Top 200. As can be seen from Tables 2 and 3, the difference between the TeslaT4 and Atlas 300V in recall and mAP in feature retrieval is less than 0.1%. Therefore, the feature results from the TeslaT4 and Atlas 300V can be mixed for feature retrieval.

Table 2. Statistics of successful feature extraction.

Platform	Dataset	Number of Images	Number of Feature	Feature Extraction Success Rate (%)
Atlas 300V	Query	41,788	41,614	99.58
	Gallery	277,440	276,495	99.66
	Interference	101,664	97,925	96.32
TeslaT4	Query	41,788	41,613	99.58
	Gallery	277,440	276,500	99.66
	Interference	101,664	97,969	96.36

Table 3. Comparison of feature retrieval on different hardware.

Similarity Threshold	TopN	Query	Gallery	Recall (%)	mAP (%)
0.3	200	TeslaT4	TeslaT4	96.32	83.12
0.3	200	Atlas 300V	Atlas 300V	96.25	82.90
0.3	200	TeslaT4	Atlas 300V	96.28	82.99
0.3	200	Atlas 300V	TeslaT4	96.29	83.03

4.2. Decode Comparison

As can be observed from Tables 4–6, compared with the Atlas 300V hard and CPU soft decoding, the TeslaT4 hard decoded more than 99% of the image with a decoding pixel maximum difference of +1/−1. The decoding success rate of software decoding is obviously higher than that of hardware decoding. Compared with Atlas 300V hard decoding, the proportion of TeslaT4 hard decoding images with the maximum difference of pixels over 100 is 0.17%. In addition, the test results on different hardware are shown in Figure 7, where the decoding maximum pixel difference is more than 100. Therefore, after hardware decoding fails, software decoding can be used instead, thereby improving the decoding success rate. The above decoding comparison experiment mainly uses JPG image for comparison.

Table 4. The effect of coding quality factor on image storage.

Atlas 300V			
Quality factor	Image 1 (200w pixel)	Image 2 (200w pixel)	Image 3 (200w pixel)
90	619 KB	602 KB	582 KB
95	848 KB	824 KB	799 KB
TeslaT4			
Quality factor	Image 1 (200w pixel)	Image 2 (200w pixel)	Image 3 (200w pixel)
75	350 KB	328 KB	335 KB
90	561 KB	527 KB	534 KB
95	802 KB	751 KB	763 KB

Table 5. Decoding difference. (image) indicates comparison by image, (pixel) indicates comparison by pixel.

Decode Compare	Number of Test Images	Pixel Difference 1 (Image)	Pixel Difference 1 (Pixel)
TeslaT4 hard vs. Atlas 300V hard	28w	99.82%	96.27%
TeslaT4 hard vs. CPU soft	28w	99.65%	95.36%
Atlas 300V hard Vs. CPU soft	28w	99.99%	99.98%

Table 6. Comparison of decoding abilities.

Decode Type	Number of Test Images	Decode Success Rate
TeslaT4 hard	28w	98.20%
Atlas 300V hard	28w	97.26%
CPU soft	28w	99.75%



Figure 7. Test results on different hardware, where the maximum pixel difference is more than 100. X hard indicates the hard decoding of X platform, and X det indicates the detection result of X platform.

4.3. Feature Comparison

As can be seen from Table 7, the scaling mode in the pre-processing process will affect the detection results, resulting in inconsistent features. In addition, the features of small target images are more sensitive to position than those of randomly selected images. From the average of feature similarity, the feature similarity between the Atlas 300V and TeslaT4 exceeds 99%. As can be seen from Tables 8 and 9, the difference between the TeslaT4 and Atlas 300V in capturing rate and accuracy of selection is within 0.1%. Additionally, the difference between the TeslaT4 and Atlas 300V in correlation accuracy and target association rate is within 0.1%. Capture rate (%) indicates the percentage of successful captures of an image with the best quality during the appearance of an object in a video. Accuracy of selection (%) represents the percentage of the capture that is the most desirable target.

Table 7. Feature similarity on different hardware. A total of 10w pictures were randomly selected for testing, respectively.

Test Type	Randomly Selected (Hard Scaling)	Small Target (Hard Scaling)	Small Target (Soft Scaling)
Similarity ≥ 0.900	99.96594	98.22877	99.02169
Similarity ≥ 0.950	99.81309	98.04452	98.89877
Similarity ≥ 0.970	99.44975	97.93458	98.80734
Similarity ≥ 0.990	96.61813	97.82261	98.72505
Similarity ≥ 0.995	89.34083	96.43209	97.61570
Max	0.99995	0.99999	0.99999
Min	0.62485	0.25368	0.25638
Mean	0.99731	0.99121	0.99537
Var	0.00003	0.00440	0.00214

Table 8. Video stream snapshot statistics. A total of 60 videos using H264 and MPEG4 encoding were tested, with resolutions ranging from 1080p to 4K and video frame rates of 25 frames per second. We sampled the video frame rate from 25 frames per second to 8 frames per second.

Type	Platform	Mark Target	Capture Rate (%)	Accuracy of Selection (%)
Rider	TeslaT4	7207	89.45	69.99
	Atlas 300V	7207	89.14	70.05
Non-motor vehicle	TeslaT4	6011	90.75	76.63
	Atlas 300V	6011	90.84	75.99

Table 9. Video stream target association statistics. A total of 60 videos using H264 and MPEG4 encoding were tested, with resolutions ranging from 1080p to 4K and video frame rates of 25 frames per second. We sampled the video frame rate from 25 frames per second to 8 frames per second.

Type	Platform	Correlation Label Quantity	Correlation Accuracy (%)	Target Association Rate (%)
Rider-non-motor vehicle	TeslaT4	3181	91.90	81.77
	Atlas 300V	3177	91.58	81.74

5. Discussion

In order to further analyze the influence of the detection results on the feature quality, we calculated the results of each pixel deviation of the detection coordinate, respectively. The coordinate deviation of the detection result is calculated using Formula (2). Then, the statistical results of feature similarity of coordinate frame deviation 0, (0,10], (10,30], (30,60), and 60+ were analyzed. As can be seen from Tables 10–12, the pixel deviation of the detection results of the Atlas 300V and TeslaT4 is mainly concentrated within 10. From the average value of feature similarity, the feature similarity between the Atlas 300V and TeslaT4 exceeds 99%.

$$\sum_{i=0}^N |x_i^1 - x_i^2| + |y_i^1 - y_i^2| \quad (2)$$

where x, y are coordinate points. N is the number of coordinate points.

Table 10. A randomly selected 10w test set is calculated according to the coordinate deviation of detection results.

Test Type	0	(0,10]	(10,30]	(30,60)	60+
Quantity proportion	0.24669	0.72476	0.02119	0.00458	0.00275
Max	0.99995	0.99995	0.99949	0.99793	0.99719
Min	0.95837	0.89377	0.75044	0.62485	0.68105
Mean	0.99891	0.99730	0.98648	0.97940	0.97181
Var	0.00000	0.00001	0.00031	0.00111	0.00160
Similarity ≥ 0.900	1.00000	0.99997	0.99340	0.98095	0.96507
Similarity ≥ 0.950	1.00000	0.99949	0.95879	0.93714	0.87619
Similarity ≥ 0.970	0.99989	0.99791	0.88092	0.83047	0.75873
Similarity ≥ 0.990	0.99833	0.97143	0.61392	0.45333	0.26666
Similarity ≥ 0.995	0.98353	0.88584	0.38030	0.11809	0.05079

Table 11. The test set of 10w small targets is calculated according to the coordinate deviation of detection results (hard scaling).

Test Type	0	(0,10]	(10,30]	(30,60)	60+
Quantity proportion	0.26178	0.67103	0.06462	0.00163	0.00091
Max	0.99999	0.99999	0.99999	0.99998	0.99981
Min	0.26454	0.25428	0.25368	0.26778	0.27480
Mean	0.99517	0.99143	0.98288	0.90586	0.43359
Var	0.00206	0.00427	0.00918	0.03914	0.03834
Similarity ≥ 0.900	0.98926	0.98260	0.96802	0.80124	0.08888
Similarity ≥ 0.950	0.98751	0.98091	0.96471	0.77639	0.08888
Similarity ≥ 0.970	0.98615	0.97999	0.96298	0.77018	0.08888
Similarity ≥ 0.990	0.98499	0.97894	0.96125	0.77018	0.08888
Similarity ≥ 0.995	0.97095	0.96548	0.94408	0.72049	0.07777

Table 12. The test set of 10w small targets is calculated according to the coordinate deviation of detection results (soft scaling).

Test Type	0	(0,10]	(10,30]	(30,60)	60+
Quantity proportion	0.56585	0.39010	0.04351	0.00025	0.00028
Max	0.99999	0.99999	0.99998	0.99997	0.99548
Min	0.26454	0.26694	0.25638	0.31350	0.27480
Mean	0.99525	0.99639	0.99225	0.92336	0.39257
Var	0.00205	0.00176	0.00394	0.04362	0.01679
Similarity ≥ 0.900	0.98965	0.99242	0.98459	0.88000	0.03571
Similarity ≥ 0.950	0.98798	0.99182	0.98342	0.88000	0.03571
Similarity ≥ 0.970	0.98678	0.99132	0.98248	0.88000	0.03571
Similarity ≥ 0.990	0.98556	0.99104	0.98202	0.88000	0.03571
Similarity ≥ 0.995	0.96385	0.45969	0.14557	0.02597	0.00000

6. Conclusions

In this article, we realize an image-retrieval method based on cross-hardware platform features, aiming to prove that the features of heterogeneous hardware platforms can be mixed, in which the Huawei Atlas 300V and NVIDIA TeslaT4 are used for experiments. First, we compare the decoding differences of heterogeneous hardware, and use CPU software decoding to help hardware decoding improve the decoding success rate. Then, we compared the difference between the Atlas 300V and TeslaT4 chip architectures and test the differences between the two platforms' features by calculating feature similarity. In addition, the scaling mode in the pre-processing process is also compared to further analyze the factors affecting the feature consistency. Next, the consistency of capture and correlation based on video structure is verified. Finally, the experimental results reveal that the feature results from the TeslaT4 and Atlas 300V can be mixed for image retrieval based on cross-hardware platform features. Consequently, cross-platform image retrieval with low error is realized. In the future, we will introduce more chips for comparison, such as

the AX650N, MLU370, DCU Z100, and further study the impact of model quantization on performance and accuracy, such as FP8.

Author Contributions: Conceptualization, H.S. and J.Y.; methodology, F.W.; software, H.S.; validation, H.S., J.Y., and F.W.; formal analysis, H.S.; investigation, J.Y.; resources, H.S.; data curation, F.W.; writing—original draft preparation, F.W.; writing—review and editing, H.S.; visualization, H.S.; supervision, J.Y.; project administration, J.Y.; funding acquisition, J.Y. and H.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All relevant data presented in the article are stored according to institutional requirements and, as such, are not available online. However, all data used in this manuscript can be made available upon request to the authors.

Acknowledgments: The authors sincerely thank Zhejiang Dahua Technology Co., Ltd. for providing product data and supporting the consistency verification experiment of the Atlas 300V and TeslaT4. In addition, we are particularly grateful to Huang Peng for his support and assistance in this project.

Conflicts of Interest: Author Hao Su was employed by the company Zhejiang Dahua Technology CO., LTD. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Xue, R.; Han, D.; Yan, M.; Zou, M.; Yang, X.; Wang, D.; Li, W.; Tang, Z.; Kim, J.; Ye, X.; et al. HiHGNN: Accelerating HGNNs Through Parallelism and Data Reusability Exploitation. *IEEE Trans. Parallel Distrib. Syst.* **2024**, *35*, 1122–1138. [CrossRef]
2. Fan, Z.; Hu, W.; Liu, F.; Xu, D.; Guo, H.; He, Y.; Peng, M. A Hardware Design Framework for Computer Vision Models Based on Reconfigurable Devices. *ACM Trans. Reconfigurable Technol. Syst.* **2024**, *17*, 2. [CrossRef]
3. Huang, B.-Y.; Lyubomirsky, S.; Li, Y.; He, M.; Smith, G.H.; Tambe, T.; Gaonkar, A.; Canumalla, V.; Cheung, A.; Wei, G.-Y.; et al. Application-level Validation of Accelerator Designs Using a Formal Software/Hardware Interface. *ACM Trans. Des. Autom. Electron. Syst.* **2024**, *29*, 35. [CrossRef]
4. Surianarayanan, C.; Lawrence, J.J.; Chelliah, P.R.; Prakash, E.; Hewage, C. A Survey on Optimization Techniques for Edge Artificial Intelligence (AI). *Sensors* **2023**, *23*, 1279. [CrossRef] [PubMed]
5. Mummidi, C.S.; Ferreira, V.C.; Srinivasan, S.; Kundu, S. Highly Efficient Self-checking Matrix Multiplication on Tiled AMX Accelerators. *ACM Trans. Archit. Code Optim.* **2024**, *21*, 21. [CrossRef]
6. Santos, F.F.D.; Carro, L.; Vella, F.; Rech, P. Assessing the Impact of Compiler Optimizations on GPUs Reliability. *ACM Trans. Archit. Code Optim.* **2024**, *21*, 26. [CrossRef]
7. T4-Tensor-Core-Product-Brief. Available online: <https://www.nvidia.cn/content/dam/en-zz/Solutions/Data-Center/tesla-t4/t4-tensor-core-product-brief.pdf> (accessed on 6 May 2024).
8. Inference-Whitepaper. Available online: <https://www.nvidia.com/en-us/lp/ai/inference-whitepaper/> (accessed on 6 May 2024).
9. NVIDIA H100 TENSOR CORE GPU. Available online: <https://images.nvidia.cn/aem-dam/en-zz/Solutions/data-center/h100/nvidia-h100-datasheet-nvidia-a4-2287922-r7-zhCN.pdf> (accessed on 6 May 2024).
10. NVIDIA A100 TENSOR CORE GPU. Available online: <https://www.nvidia.cn/data-center/a100/> (accessed on 6 May 2024).
11. Miliadis, P.; Theodoropoulos, D.; Pnevmatikatos, D.; Koziris, N. Architectural Support for Sharing, Isolating and Virtualizing FPGA Resources. *ACM Trans. Archit. Code Optim.* **2024**, *21*, 33. [CrossRef]
12. Xie, K.; Lu, Y.; He, X.; Yi, D.; Dong, H.; Chen, Y. Winols: A Large-Tiling Sparse Winograd CNN Accelerator on FPGAs. *ACM Trans. Archit. Code Optim.* **2024**, *21*, 31. [CrossRef]
13. Jouppi, N.P.; Young, C.; Patil, N.; Patterson, D.; Agrawal, G.; Bajwa, R.; Bates, S.; Bhatia, S.; Boden, N.; Borchers, A.; et al. In-Datcenter Performance Analysis of a Tensor Processing Unit. In Proceedings of the ISCA '17: The 44th Annual International Symposium on Computer Architecture, Toronto, ON, Canada, 24–28 June 2017.
14. Apple a13. Available online: https://en.wikipedia.org/wiki/Apple_A13 (accessed on 6 May 2024).
15. Snapdragon 865. Available online: <https://www.qualcomm.com/products/snapdragon-865-5g-mobile-platform> (accessed on 6 May 2024).
16. KL730 AI Soc. Available online: <https://www.kneron.com/cn/page/soc/> (accessed on 6 May 2024).

17. Atlas 300V Video Analysis Card User Guide. Available online: <https://support.huawei.com/enterprise/en/doc/EDOC1100285915/3965035e/product-features?idPath=23710424%7C251366513%7C22892968%7C252309139%7C256209253/> (accessed on 6 May 2024).
18. Liao, H.; Tu, J.; Xia, J.; Liu, H.; Zhou, X.; Yuan, H.; Hu, Y. Ascend: A Scalable and Unified Architecture for Ubiquitous Deep Neural Network Computing: Industry Track Paper. In Proceedings of the 2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA), Seoul, Republic of Korea, 27 February–3 March 2021; pp. 789–801.
19. Deng, W.; Guan, K.; Shi, Y.; Liu, T.; Yuan, H. Research on Performance of Deep Learning Computing Card. *Manuf. Upgrad. Today Chin.* **2023**, *7*, 103–107.
20. Lu, W.; Zhang, F.; He, Y.-X.; Chen, Y.; Zhai, J.; Du, X. Performance Evaluation and Optimization of Huawei Centeng Neural Net-work Accelerator. *Chin. J. Comput.* **2022**, *45*, 1618–1637. [[CrossRef](#)]
21. Kum, S.; Oh, S.; Yeom, J.; Moon, J. Optimization of Edge Resources for Deep Learning Application with Batch and Model Management. *Sensors* **2022**, *22*, 6717. [[CrossRef](#)]
22. Goel, A.; Tung, C.; Hu, X.; Thiruvathukal, G.K.; Davis, J.C.; Lu, Y.-H. Efficient Computer Vision on Edge Devices with Pipeline-Parallel Hierarchical Neural Networks. In Proceedings of the 2022 27th Asia and South Pacific Design Automation Conference (ASP-DAC), Taipei, Taiwan, 17–20 January 2022.
23. Wu, Q.; Shen, Y.; Zhang, M. Heterogeneous Computing and Applications in Deep Learning: A Survey. In Proceedings of the 5th International Conference on Computer Science and Software Engineering, Guilin, China, 21–23 October 2022; pp. 383–387.
24. Zhang, X.; Hao, C.; Zhou, P.; Jones, A.; Hu, J. H2H: Heterogeneous Model to Heterogeneous System Mapping with Computation and Communication Awareness. In Proceedings of the Proceedings of the 59th ACM/IEEE Design Automation Conference, San Francisco, CA, USA, 10–14 July 2022; pp. 601–606.
25. Zhuang, J.; Huang, X.; Yang, Y.; Chen, J.; Yu, Y.; Gao, W.; Li, G.; Chen, J.; Zhang, T. OpenMedIA: Open-Source Medical Image Analysis Toolbox and Benchmark under Heterogeneous AI Computing Platforms. *arXiv* **2022**, arXiv:2208.05616.
26. Prerna, M.; Kumar, S.; Chaube, M.K. An Efficient Image Retrieval Method Using Fused Heterogeneous Feature. *Pattern Recognit. Image Anal.* **2020**, *30*, 674–690. [[CrossRef](#)]
27. Alsmadi, M.K. Content-Based Image Retrieval Using Color, Shape and Texture Descriptors and Features. *Arab. J. Sci. Eng.* **2020**, *45*, 3317–3330. [[CrossRef](#)]
28. Chhabra, P.; Garg, N.K.; Kumar, M. Content-based image retrieval system using ORB and SIFT features. *Neural Comput. Appl.* **2020**, *32*, 2725–2733. [[CrossRef](#)]
29. Öztürk, Ş. Stacked auto-encoder based tagging with deep features for content-based medical image retrieval. *Expert Syst. Appl.* **2020**, *161*, 113693. [[CrossRef](#)]
30. Li, X.; Yang, J.; Ma, J. Recent developments of content-based image retrieval (CBIR). *Neurocomputing* **2021**, *452*, 675–689. [[CrossRef](#)]
31. Putzu, L.; Piras, L.; Giacinto, G. Convolutional neural networks for relevance feedback in content based image retrieval. *Multimed. Tools Appl.* **2020**, *79*, 26995–27021. [[CrossRef](#)]
32. Muthusami, R. Content-based image retrieval using the heterogeneous features. In Proceedings of the 2010 International Conference on Signal and Image Processing, Chennai, India, 15–17 December 2010; pp. 494–497.
33. Wu, J.; He, Y.; Guo, X.; Zhang, Y.; Zhao, N. Heterogeneous Manifold Ranking for Image Retrieval. *IEEE Access* **2017**, *5*, 16871–16884. [[CrossRef](#)]
34. Wang, L.; Wang, H. Improving feature matching strategies for efficient image retrieval. *Signal Process. Image Commun.* **2017**, *53*, 86–94. [[CrossRef](#)]
35. Yu, Y.; Yang, L.; Zhou, H.; Zhao, R.; Li, Y.; Tong, H.; Miao, X. In-Memory Search for Highly Efficient Image Retrieval. *Adv. Intell. Syst.* **2023**, *5*, 2200268. [[CrossRef](#)]
36. Rani, L.N.; Yuhandri, Y. Similarity Measurement on Logo Image Using CBIR (Content Base Image Retrieval) and CNN ResNet-18 Architecture. In Proceedings of the 2023 International Conference on Computer Science, Information Technology and Engineering (ICCoSITE), Jakarta, Indonesia, 16 February 2023.
37. Xin, J.; Ye, F.; Xia, Y.; Luo, Y.; Chen, X. A New Remote Sensing Image Retrieval Method Based on CNN and YOLO. *J. Internet Technol.* **2023**, *24*, 233–242. [[CrossRef](#)]
38. Lee, G.-W.; Maeng, J.-H.; Song, S. Content based Image Retrieval Method that Combining CNN based Image Features and Object Recognition Information. *J. Korean Inst. Inf. Technol.* **2022**, *20*, 31–37. [[CrossRef](#)]
39. Wang, L.; Qian, X.; Zhang, Y.; Shen, J.; Cao, X. Enhancing Sketch-Based Image Retrieval by CNN Semantic Re-ranking. *IEEE Trans. Cybern.* **2020**, *50*, 3330–3342. [[CrossRef](#)]
40. Zhang, N.; Liu, Y.; Li, Z.; Xiang, J.; Pan, R. Fabric image retrieval based on multi-modal feature fusion. *Signal Image Video Process.* **2024**, *18*, 2207–2217. [[CrossRef](#)]
41. Zhan, Z.; Zhou, G.; Yang, X. A Method of Hierarchical Image Retrieval for Real-Time Photogrammetry Based on Multiple Features. *IEEE Access* **2020**, *8*, 21524–21533. [[CrossRef](#)]
42. Ye, F.; Xiao, H.; Zhao, X.; Dong, M.; Luo, W.; Min, W. Remote Sensing Image Retrieval Using Convolutional Neural Network Features and Weighted Distance. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 1535–1539. [[CrossRef](#)]
43. Fan, Z.; Guan, Y. Secure Image Retrieval Based on Deep CNN Features in Cloud Computing. In Proceedings of the 2022 3rd International Conference on Pattern Recognition and Machine Learning (PRML), Chengdu, China, 22–24 July 2022; pp. 186–192.

44. Lv, Y.; Wang, C.; Yuan, W.; Qian, X.; Yang, W.; Zhao, W. Transformer-Based Distillation Hash Learning for Image Retrieval. *Electronics* **2022**, *11*, 2810. [CrossRef]
45. Li, Y.; Guan, C.; Gao, J. TsP-Tran: Two-Stage Pure Transformer for Multi-Label Image Retrieval. In Proceedings of the 2023 ACM International Conference on Multimedia Retrieval, Thessaloniki, Greece, 12–15 June 2023; pp. 425–433.
46. Li, X.; Yu, J.; Jiang, S.; Lu, H.; Li, Z. MSViT: Training Multiscale Vision Transformers for Image Retrieval. *Trans. Multi.* **2023**, *26*, 2809–2823. [CrossRef]
47. Hu, Y.; Liu, Y.; Liu, Z. A Survey on Convolutional Neural Network Accelerators: GPU, FPGA and ASIC. In Proceedings of the 2022 14th International Conference on Computer Research and Development (ICCRD), Shenzhen, China, 7–9 January 2022; pp. 100–107.
48. Zhang, H.; Subbian, D.; Lakshminarayanan, G.; Ko, S.-B. Application-Specific and Reconfigurable AI Accelerator. In *Artificial Intelligence and Hardware Accelerators*; Mishra, A., Cha, J., Park, H., Kim, S., Eds.; Springer International Publishing: Cham, Switzerland, 2023; pp. 183–223.
49. Kalapothas, S.; Flami, G.; Kitsos, P. Efficient Edge-AI Application Deployment for FPGAs. *Information* **2022**, *13*, 279. [CrossRef]
50. Nechi, A.; Groth, L.; Mulhem, S.; Merchant, F.; Buchty, R.; Berekovic, M. FPGA-based Deep Learning Inference Accelerators: Where Are We Standing? *ACM Trans. Reconfigurable Technol. Syst.* **2023**, *16*, 60. [CrossRef]
51. Wu, R.; Guo, X.; Du, J.; Li, J. Accelerating Neural Network Inference on FPGA-Based Platforms—A Survey. *Electronics* **2021**, *10*, 1025. [CrossRef]
52. Machupalli, R.; Hossain, M.; Mandal, M. Review of ASIC accelerators for deep neural network. *Microprocess. Microsyst.* **2022**, *89*, 104441. [CrossRef]
53. Tang, Y.; Zhou, P.; Zhang, W.; Hu, H.; Yang, Q.; Xiang, H.; Liu, T.; Shan, J.; Huang, R.; Zhao, C.; et al. Exploring Performance and Cost Optimization with ASIC-Based CXL Memory. In Proceedings of the Nineteenth European Conference on Computer Systems, Athens, Greece, 22–25 April 2024; pp. 818–833.
54. CANN6.0.1. Available online: https://www.hiascend.com/document/detail/en/canncommercial/601/inferapplicationdev/aclythondevg/aclythondevg_01_0309.html (accessed on 6 May 2024).
55. CANN. Available online: <https://www.hiascend.com/en/software/cann> (accessed on 6 May 2024).
56. Recommended 8-Bit YUV Formats for Video Rendering. Available online: <https://learn.microsoft.com/en-us/windows/win32/medfound/recommended-8-bit-yuv-formats-for-video-rendering> (accessed on 30 May 2024).
57. NVIDIA-Turing-Architecture-Whitepaper. Available online: <https://images.nvidia.cn/aem-dam/en-zz/Solutions/design-visualization/technologies/turing-architecture/NVIDIA-Turing-Architecture-Whitepaper.pdf> (accessed on 6 May 2024).
58. NVIDIA GPU Architecture: From Pascal to Turing to Ampere. Available online: <https://wolfadvancedtechnology.com/articles/nvidia-gpu-architecture> (accessed on 20 May 2024).
59. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
60. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
61. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv* **2022**, arXiv:2207.02696.
62. Terven, J.; Cordova-Esparza, D. A Comprehensive Review of YOLO: From YOLOv1 to YOLOv8 and Beyond. *arXiv* **2023**, arXiv:2304.00501.
63. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
64. Xie, S.; Girshick, R.; Dollar, P.; Tu, Z.; He, K. Aggregated Residual Transformations for Deep Neural Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
65. Howard, A.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
66. Ma, N.; Zhang, X.; Zheng, H.-T.; Sun, J. ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design. In *Computer Vision—ECCV 2018, Lecture Notes in Computer Science*; Springer: Cham, Switzerland, 2018; pp. 122–138.
67. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
68. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. *Proc. AAAI Conf. Artif. Intell.* **2017**, *31*, 4278–4284. [CrossRef]
69. Atlas 300V Pro Video Analysis Card User Guide. Available online: <https://support.huawei.com/enterprise/en/doc/EDOC1100209002> (accessed on 6 May 2024).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.