

Face Recognition Product Suite



From Cloud to Edge, Paravision's comprehensive face recognition product suite offers all of the tools necessary to develop and deploy mission-critical biometric identification and authentication solutions across a wide range of applications, from identity verification to physical access control to air travel and government services.

Paravision face recognition has been optimized for leading chipsets, from Intel CPUs to NVIDIA GPUs to Edge SoCs, Apple silicon, and the broader ARM ecosystem. In addition, Paravision face recognition supports leading operating systems including Windows, Linux, iOS, and Android, and can be deployed at any level, from SDK to API- driven Docker containers.

Paravision has been repeatedly recognized by NIST FRTE as a top global provider and the most accurate technology provider in the U.S., UK, and EU across use cases, including 1:1 verification and 1:N identification, including multi-million record databases, performance by age, performance with face masks, and the full range of image quality characteristics, from passport-quality to fully unconstrained images.

In short, Paravision powers its partners with world-class face recognition technology in a way that fits their technical expertise and deployment goals, supporting transformative solutions for the next generation identity, security, efficiency, and user experiences.

Face Recognition SDKs - From Cloud to Edge

For developers looking to deeply integrate Paravision face recognition into software applications, solutions, and services, Paravision offers a rich, multi-platform series of face recognition SDKs. For cloud, on-premises, and desktop applications, Paravision offers C++ and Python SDKs for Windows and Linux. For high-speed, high-precision Edge AI applications, Paravision offers its SDKs for Ambarella, Hailo, and Rockchip. And for mobile- or kiosk-based application development, Paravision offers its SDK for Android and iOS. All of these SDKs enable full face recognition pipeline functionality, including face detection, face landmark identification, image quality analysis, template (embedding) generation, and 1:1 or 1:N matching.

To implement the optimal systems architecture for a given use case, developers can pick and choose aspects of Paravision's SDKs across platforms. For instance, face detection and image quality analysis could be used for mobile registration, passing cropped, quality-checked faces to backend systems for registration. Paravision offers fully interoperable templates (embeddings), which can be created on the server and distributed to networked Edge or mobile devices, which could then be used for 1:N matching. Alternatively, the full face recognition pipeline could be deployed server-side or on the Edge in each of these cases-whatever is appropriate for the application needs.

Face Recognition Containers - Rapidly Deployable, Massively Scalable

For developers looking to access Paravision face recognition functionality at a higher level, Paravision offers a series of Docker container-based face recognition engines. With all required face recognition capability fully integrated, these engines can be rapidly integrated into many standard cloud- or on-premises systems architectures, and are accessible by either REST or gRPC (and associated client libraries).

Paravision Search – Scalable, Elastic, Enterprise-Grade Face Recognition

Paravision Search takes Paravision Face Recognition, which has been available until now as an SDK or discrete docker-based matcher, and integrates it into a true enterprise-grade search system. This includes massive scalability, supporting both very large databases and extreme request concurrency. It also includes native elasticity, allowing Paravision partners to adjust compute resources to meet real-time demands for peak traffic or quiet periods. See Paravision Search datasheet for more information.

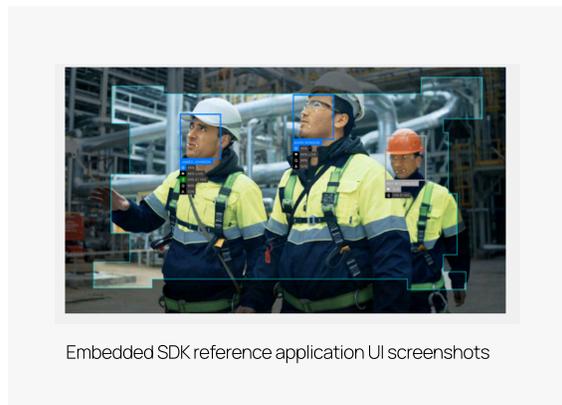
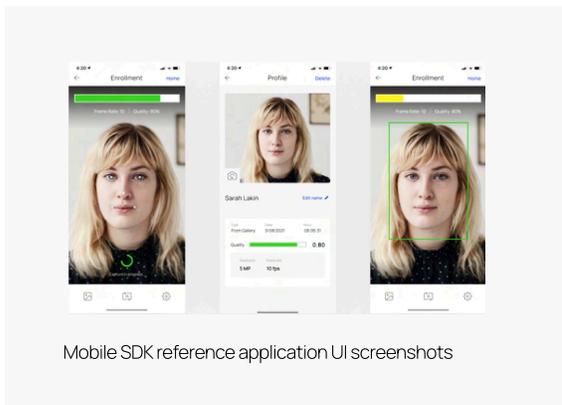
Supported Computing Environments

Paravision supports a wide range of computing environments. Paravision proudly partners with leaders from Silicon Valley and around the world to deliver outstanding price, performance, availability, and support:

Intel	NVIDIA	Ambarella	Hailo	Rockchip	ARM	Apple silicon
						
Supported Computer Vision Framework						
						

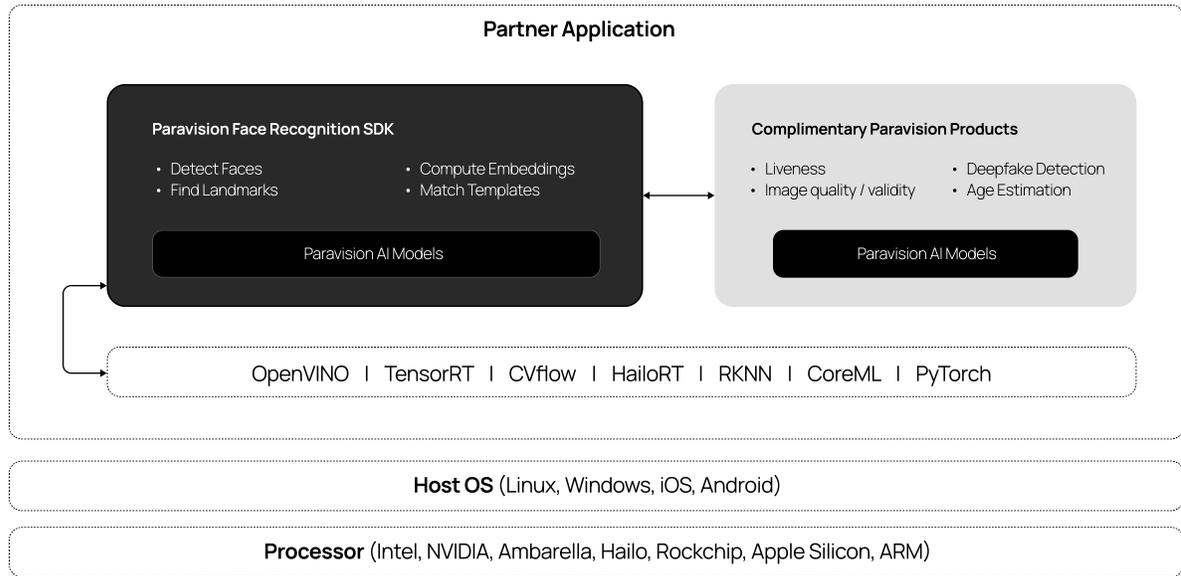
Implementation Support

To speed up time-to-market, reduce development risk, and otherwise optimize partner integration efforts, Paravision offers a range of demonstration applications, user interface examples, and sample code. Sampler user interface graphics for mobile and embedded applications can be seen here.

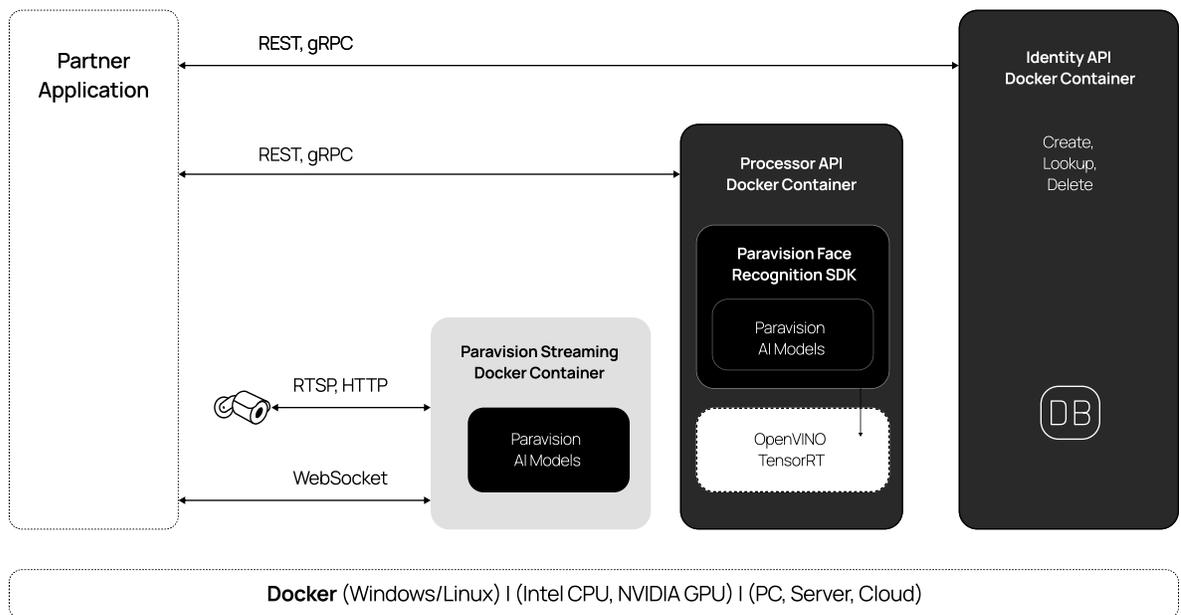


System Architectures

FACE RECOGNITION SDKS



FACE RECOGNITION CONTAINERS



Technical Specifications

FACE RECOGNITION SDKS

	Mobile	Embedded	Desktop/Server
Programming languages	Swift - iOS Kotlin - Android	C++ / C Wrapper	C++ / Python Wrapper
Supported operating systems	iOS 13+ Android 8.0+ (API ver 26)	Linux for Ambarella CVflow Rockchip Hailo	Windows 10+ Windows Server 2019 Datacenter Linux: Ubuntu 20.04
Supported hardware platforms & computer vision frameworks	Apple Silicon (CoreML) ARM (PyTorch)	Ambarella CV25/CV22 (CVflow) Rockchip (RKNN) Hailo (HailoRT)	Intel (OpenVINO) NVIDIA (TensorRT)
Supported functions	Face Detection <ul style="list-style-type: none"> • Face detection / bounding box detection • Face landmarks identification • Image quality analysis Face Recognition : <ul style="list-style-type: none"> • Template (embedding) generation • Toolsets enabling 1:1 verification and 1:N identification 		

FACE RECOGNITION CONTAINERS

Deployment method	Docker container, supporting on-premises or cloud-based computing
Supported operating systems	Windows 10+, Windows Server 2019 Datacenter, Linux: Ubuntu 20.04
Supported compute platforms (Computer vision frameworks)	Intel CPU (OpenVINO) NVIDIA GPU (TensorRT)
Face recognition engine APIs	REST gRPC, with Clients supporting Python, C++, C#, Node.JS, GoLang, Java, Ruby
Supported functions	Face Detection: <ul style="list-style-type: none"> • Face detection / bounding box detection • Face landmarks identification • Image quality analysis Face Recognition: <ul style="list-style-type: none"> • Template (embedding) generation • 1:1 verification • 1:N identification Additional Processor Docker Functions: <ul style="list-style-type: none"> • Liveness • Age Estimation • Deepfake Detection