VisionLabs
MACHINES CAN SEE

VisionLabs LUNA PLATFORM

System Overview

VisionLabs B.V.

Keizersgracht 311, 1016 EE, Amsterdam, the Netherlands

+31 20 369 04 93

info@visionlabs.ai

www.visionlabs.ai

Table of contents

Introduction

Welcome to LUNA PLATFORM! This document provides a high-level description of the service, the tasks it solves and the methods involved, as well as terminology, system design concepts and rationale.

The Overview section covers common tasks solved by the platform. LUNA PLATFORM section introduces to system design and architecture. The following sections tell about various subsystems in depth. The Licensing section tell about HASP service, which used for licensing.

1 Overview

LUNA PLATFORM is a facial recognition web service.

The platform is designed to solve the following problems:

- Images loading and analysis. Analysis involves:
  - face detection,
  - facial attributes estimation (gender, age),
  - face descriptor extraction and matching (see section 2 for details);
- Facial identification and verification using descriptors;
- Efficient descriptor storage for fast searching;
- Logical grouping of descriptors;
- Logging and events notification;
- Providing graphical user interface for the features;
- Account management and access restriction.

2 LUNA PLATFORM

LUNA PLATFORM consists of several components: the API, the CORE, the Events and statistics, the UI and administration services as shown in Fig. 1:

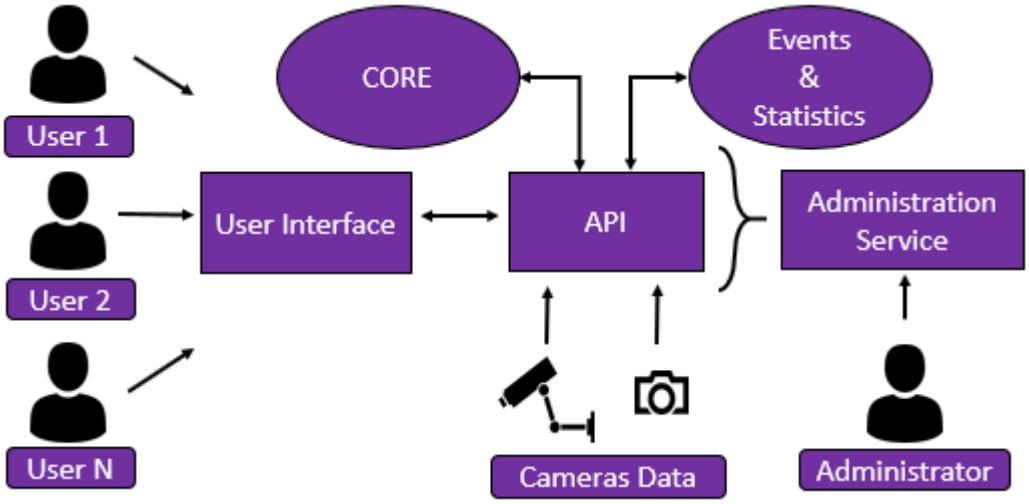

Figure 1 - LUNA PLATFORM architecture

The system is designed to work with images (either still photographs or individual video frames).

The system tries to detect all human faces it can on each submitted image. For each detected face, the system outputs a bounding box and a set of key points (landmarks) for eyes, nose and mouth. In addition, several more attributes may be estimated upon request, including age and gender.

Each detected face may be converted into a special set of unique features, called face descriptor. A descriptor requires much less storage memory in comparison with a source image. In addition, it is impossible to restore original face image form the descriptor, which is important for personal data safety reasons. This conversion process is called descriptor extraction. All extracted descriptors are automatically preserved in the system database. See Fig. 2 for details:

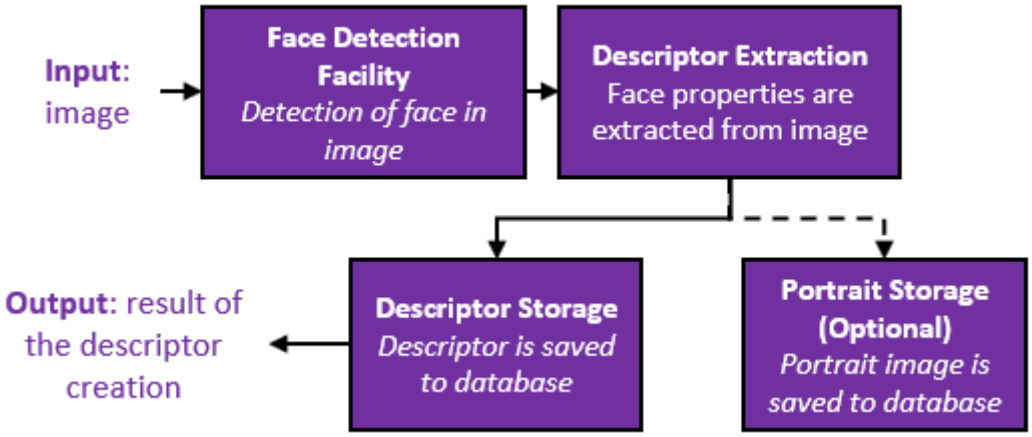

Figure 2 - Descriptor extraction pipeline

Once a descriptor is stored in the DB, the platform does not need the source image any longer. However, it can be configured to keep portrait images around for presentation and system upgrade purposes. The platform provides a customizable storage facility with multiple backing options to choose from.

A pair of descriptors may be compared to determine whether they belong to the same person or not. This comparison is called descriptor matching. Given a face descriptor, the platform allows to search the similar-looking faces in the database by matching stored descriptors.

The system implements facial verification and identification by means of matching the select descriptors. The matching pipeline is in Fig. 3:

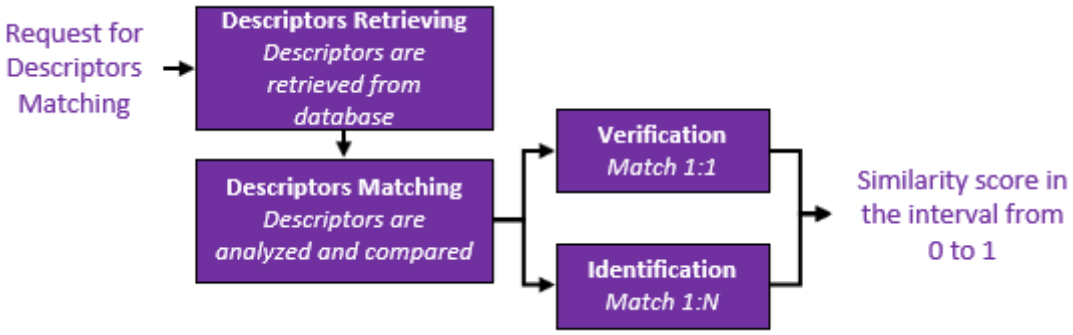

Figure 3 - Descriptor matching pipeline

The main gateway to the system is the API service. It provides a RESTful interface allowing the clients to use aforementioned detection, extraction and matching procedures. The API is performant and flexible and allows 3rd party systems and smart device integration.

The UI service implements graphical interface for human users. It builds atop of the same public API.

The Events & Statistics service logs all requests for descriptor extraction and matching. Each such request is referred to as an event. The service allows gathering various event statistics including duration, distribution in time, error to success ratio and other metrics. The service is also capable of event notification in real time via web sockets. This allows easy integration with chat bots, web sites and other applications.

The Administration service solves common maintenance tasks including user account management (creation, suspension and removal), system monitoring and upgrades.

2.1 LUNA API

The API is designed to provide restricted user access to system resources such as face descriptors and their source images (if you choose to store them) through a RESTful application interface.

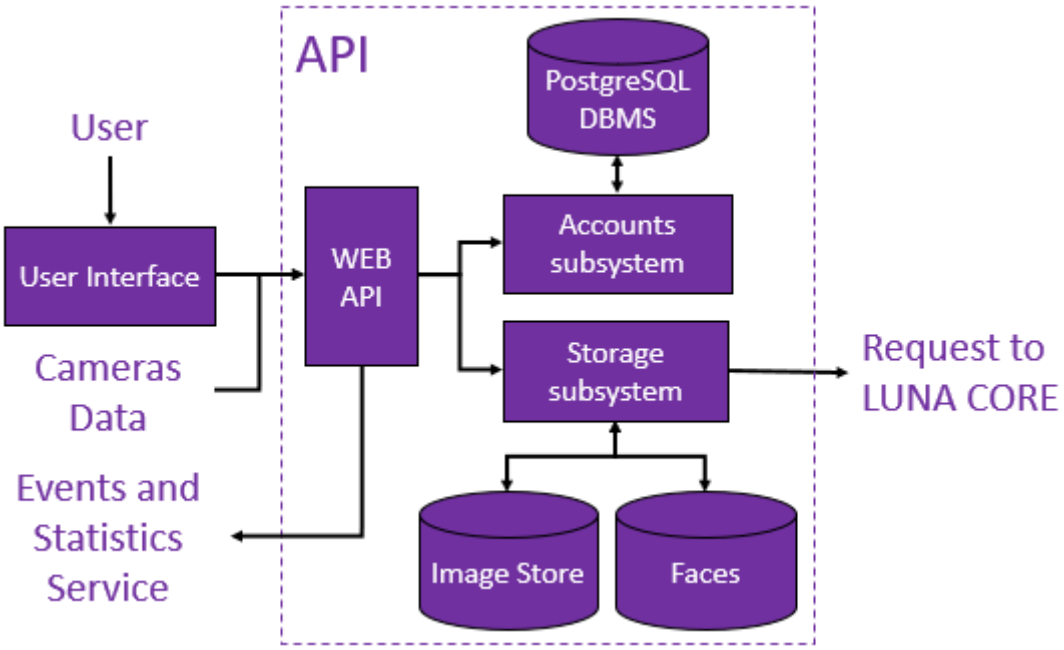

Figure 4 - LUNA API architecture

- WEB API provides access to resources according to a user account list;
- *Accounts* subsystem manages registration information and account authorization;
- *Storage* subsystem is responsible for providing access to account data once authorized;
- A relational *PostgreSQL* DBMS (or Oracle DBMS) is used to store metainformation from *Storage* and *Account* subsystems;
- Faces service stores faces, persons and lists.

Image Store is configured to preserve portrait images and other binary data directly on HDD or in S3 DB.

Storage and Accounts subsystem communicates with the CORE service to utilize its processing capabilities and access descriptor storage.

The API implements multiple authorization methods:

- Authorization using a login and password. This type is used for users and administrator authorization on their accounts;
- Authorization using an API token. The tokens are used to authorize third-party systems (or devices) with limited rights on a user account. For example, when you need to receive videos from your customer IP cameras, you should use tokens for authorization.

The API implements several ways of logical descriptor grouping:

- The system implements the notion of “persons” allowing to group together descriptors known to belong to the same real-world person;

- The system implements “lists” allowing to group together persons or individual descriptors.

For example, you can create a list of important customers, where each person has one or several face descriptors.

It is possible to attach user-defined information to lists and persons even if the information is not used by the platform. The information may be useful upon integration with 3rd party systems.

2.2 LUNA CORE

The CORE service is a scalable system that solves all the computation-intensive tasks like face detection, image normalization, descriptor extraction and matching.

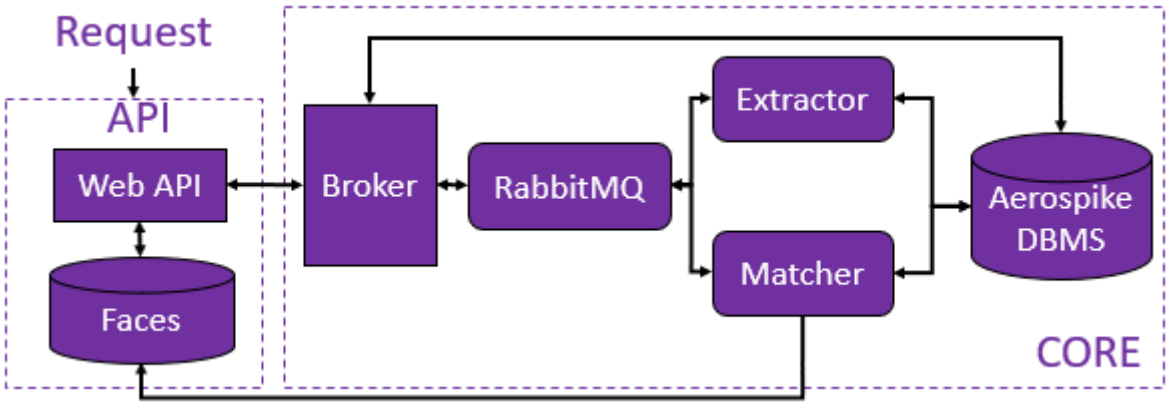

Figure 5 - LUNA CORE architecture

- *Broker* serves for communication with higher-level services;
- *RabbitMQ* service is responsible for task dispatching and returning results back;
- *Extractor* computes face descriptors and attributes from each face it finds in the input image;
- *Matcher* performs descriptors comparison for person verification or identification;
- No-SQL DBMS Aerospike is used for descriptor storage and data caching. Improves data storage and retrieval performance under high load.

Computational tasks are performed by workers. The workers are dedicated processes that take tasks and produce results. Workers do not communicate with each other and are isolated and stateless. Workers obtain tasks via network and may be distributed on several servers. There may be unlimited number of workers, and new workers may be added dynamically, hence scaling is possible. There are two workers in LUNA CORE – Extractor and Matcher.

The Broker service provides a communication channel for the API service, which dispatches API requests through a message service to deliver tasks to appropriate workers.

A separate message queue service is responsible of task dispatching to the workers and returning results back. It is possible to use industry-proven messaging systems based on AMQP protocol, like RabbitMQ.

The Broker service analyzes incoming tasks before dispatching them to workers and can optimize some kinds of them. For example, it may split large matching task into several smaller tasks processed concurrently to utilize multiple matching workers at once. Such optimizations are transparent for users and completely hidden by high level API.

2.3 Index Building and Index Search Services

LUNA PLATFORM allows to index descriptors in lists. Indexing is an additional feature for the cases when the matching speed is not sufficient. Search by indexed lists is considerably faster. The services are shown in Fig. 6.

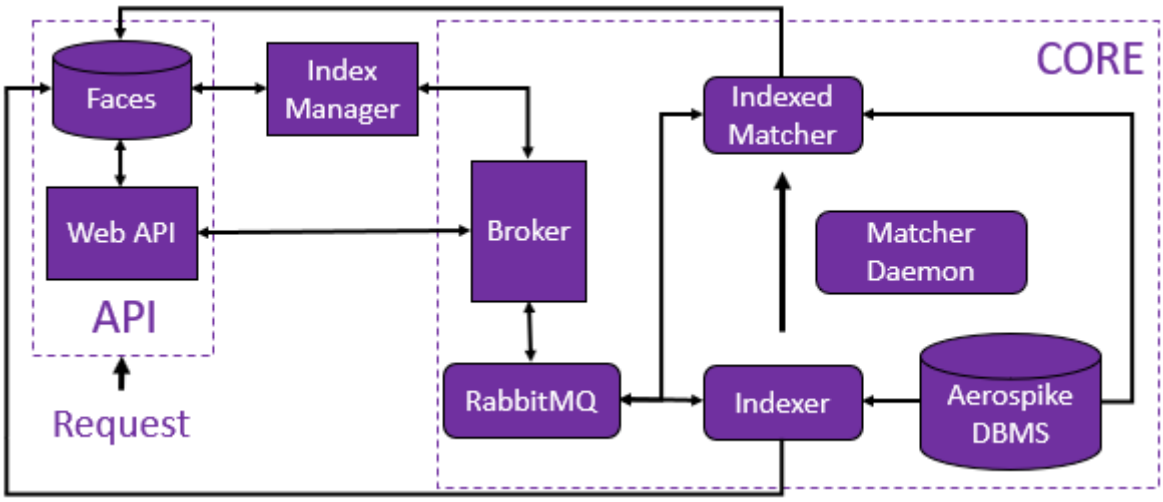

Figure 6 – Services for index building and index search

The following services are used for descriptors indexing in the list and searching in the generated index:

- Indexer creates index based on a descriptor list;
- Indexed Matcher searches by indexes;
- Matcher Daemon:
  - copies the index from the server, on which Indexer is installed, to the server for search by an index (Index Matcher),
  - restarts Indexed Matcher with a new index generation;
- Index Manager forms tasks for index building and coordinates the process of index delivery to Index Matcher servers.

2.4 Event & Statistic Service

The Event &Statistics service allows keeping track of all requests that LUNA PLATFORM processes.

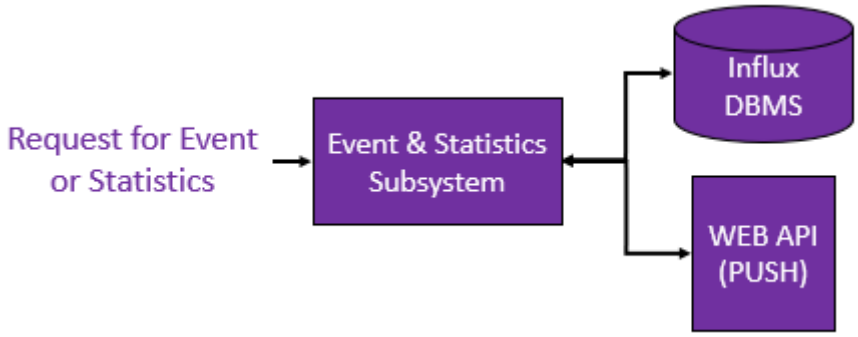

Figure 7 – Event & Statistics subsystem architecture

- Event & statistics subsystem allows to emit events and receive statistics on user requests;
- Special in-memory DBMS Influx is used to gather and store statistics.

It is possible to collect multiple metrics pre request, like:

- Time of the request execution (number of requests per seconds);
- Request status (succeeded, failed);
- Request status before failure;
- Other metrics.

The service gathers statistics in a special in-memory DBMS (Influx DB) optimized for time series storage. This allows retrieving metrics for one or multiple counters for a given period very quickly. One may visualize and brows the collected data using third party software like Grafana.

Aside from accumulating statistics, it is possible to collect events. An event happens when a descriptor is extracted or matched. Event contains the same data as LUNA API returns in response to user request that triggers the event.

The events are delivered in real time by means of web sockets making it easy to implement a custom listener no matter should it run as a native application or as a script in a web browser.

For example, you have a list of VIP guests. You can receive notifications to a smartphone about the guests arrival using an ad hoc application that has subscription to the event service.

It is possible to subscribe to multiple events at once.

2.5 UI Service

The UI service is a web application that implements a graphical interface for LUNA PLATFORM users. The interface supports all the basic actions normally available through LUNA API, such as:

- Registration and user log in;
- Image uploading and portrait gallery browsing;
- Persons and lists management;
- Person identification, verification and generic descriptor matching tasks;
- Token management.

2.6 Administration Service

The administration service is a control panel-style web application. It provides a graphical interface for solution of common administrative tasks, including:

- Viewing user accounts and their data;
- Performing face descriptor neural network model update;
- Performing garbage collection, i.e. removal of any descriptors not attached to persons or lists;
- Viewing system metrics with Grafana (a 3rd party tool).

3 Licensing

LUNA PLATFORM doesn't have its own licensing service. However, its underlying component LUNA CORE uses HASP protection suite.

Each HASP license key, independently of its type (software, hardware or network), contains a maximum number of descriptors each system instance can simultaneously allocate.

CORE workers count number of allocated descriptors. Actual number of descriptors is being compared to the limit. Due to the fact, that each query to the HASP key can take significant amount of time, the comparison is performed with some predefined frequency (say, each 100th request).

Once the system reaches 90% of the limit threshold, it starts writing a warning like this:

[Warn] [DescriptorFactory] Getting close to descriptor limit (9995001 allocated of 10000000) to the log.

Once the descriptor limit is reached/exceeded, the system writes the following error to the log:

[Error] [DescriptorFactory] Cannot allocate a descriptor. Reason: limit exceeded (10000000) and stops allocating new descriptors (and descriptor batches).

The above messages may be observed by monitoring Extractor and Matcher logs. It is generally recommended to monitor LUNA logs for warnings and errors.

CORE workers license the maximum number of allocated descriptor objects. Actual size of descriptor (face) database used by LUNA PLATFORM *may exceed* that number provided if *it will not have to keep a copy of the entire database in its worker services memory*.

LUNA Extractors allocate only individual descriptors (one at a time).

LUNA Matchers allocate:

- Descriptor batches to keep candidate lists used for matching;
- Individual descriptors to represent references for matching (one at a time).

Candidate descriptor batches may be cached by LUNA Matcher to speed up performance. Depending on cache setup, considerable amount of descriptors may present in the LUNA Matcher process memory (up to several millions).

One should carefully choose caching settings depending on actual database size and the number of Matchers (in case of network licenses).

For further details on HASP, please visit https://sentinel.gemalto.com/software-monetization/sentinel-hasp/

Appendix: version history

Date	Version	Notes
25.05.17	1	Initial release.
19.09.17	2	Added licensing methods.
20.09.17	3	Updated extensions description.
28.02.18	4	The document was brought in line with the corporate style. Updated section about administrative service.
29.05.18	5	The document was updated according to system changes.
10.07.18	6	Added descriptions for Indexer, Indexed Matcher, Matcher Daemon, Index Manager, Image Store and Faces services