Detecting Features and Text in Images

Janani Ravi
CO-FOUNDER, LOONYCORN
www.loonycorn.com

Overview

Importance of feature detection

Scale Invariant Feature Transform (SIFT) and DAISY

Histogram of Oriented Gradients (HOG)

Optical Character Recognition (OCR)

Using Images in Machine Learning

Image pre-processing
Normalization, aspect ratio etc

Covered in previous module

Image de-noising
Specialized form of pre-processing

ZCA whitening, use of filters

Feature detection
Key points and blobs

Points and regions of interest

Computation of Image descriptors
Properties associated with key points and blobs

SIFT, Daisy, Histogram of Oriented Gradients (HOG)

Application of ML algorithm
CNNs, RNNs, DNNs etc

Classification, dimensionality reduction

Using Images in Machine Learning

Image pre-processing

Normalization, aspect ratio etc

Covered in previous module

Image de-noising

Specialized form of pre-processing

ZCA whitening, use of filters

The Growing Importance of Feature Detection

Using Images in Machine Learning

Image pre-processing

Normalization, aspect ratio etc

Covered in previous module

Image de-noising

Specialized form of pre-processing

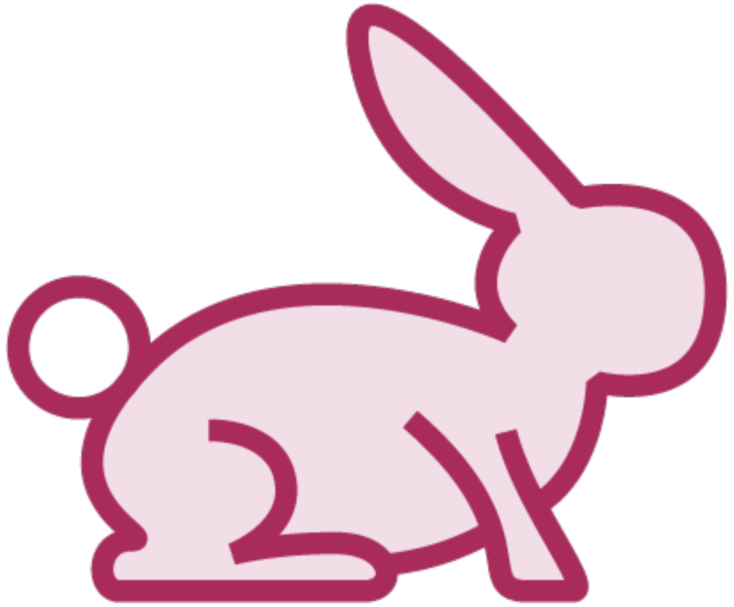
ZCA whitening, use of filters

Feature detection

Key points and blobs

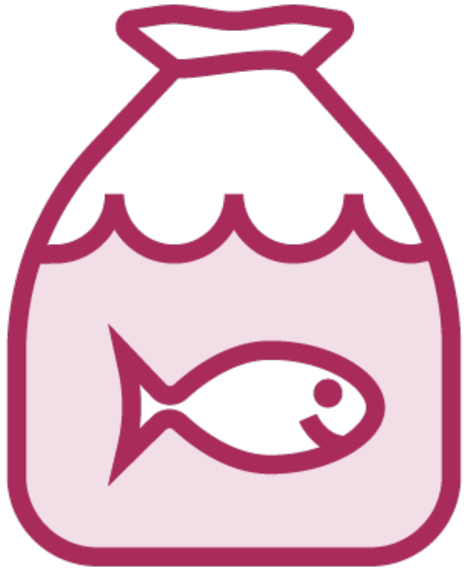
Points and regions of interest

Whales: Fish or Mammals?



Mammals

Members of the infraorder *Cetacea*

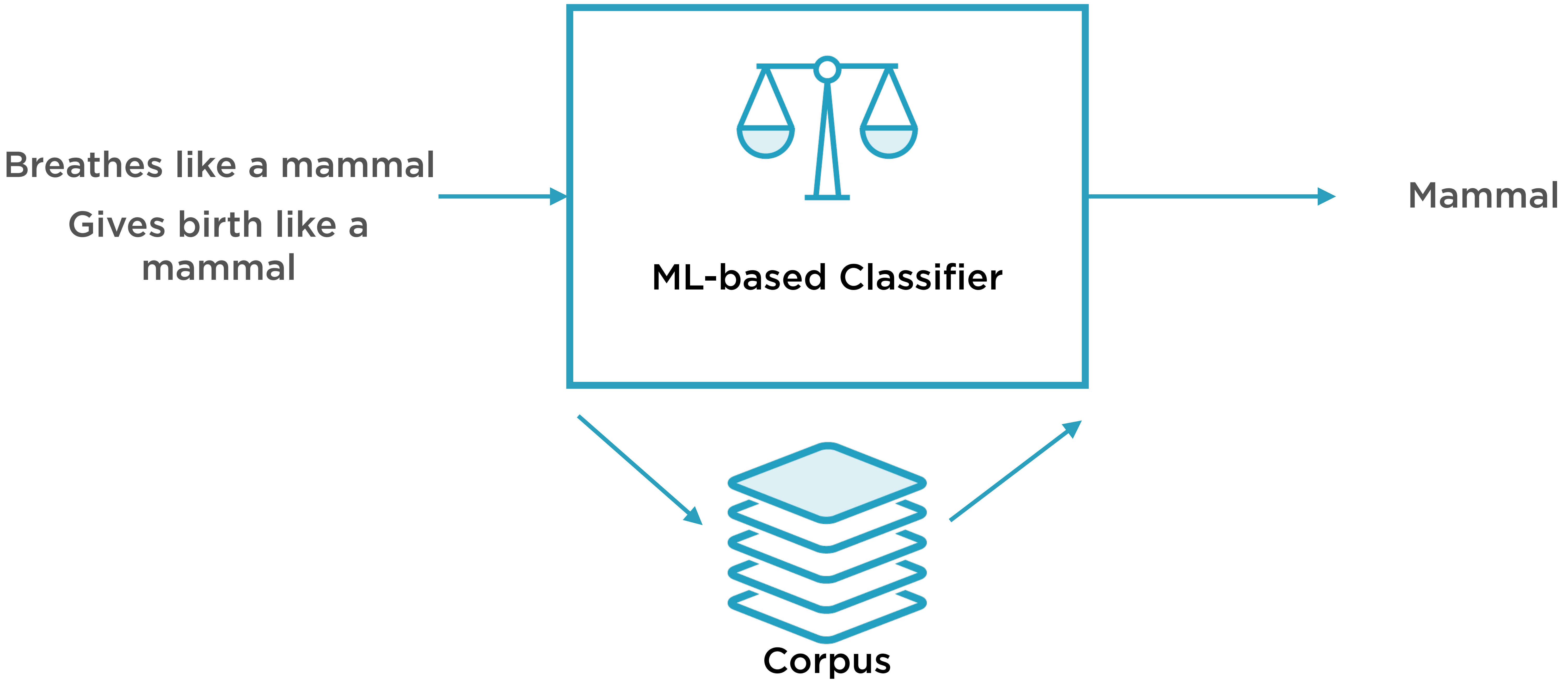


Fish

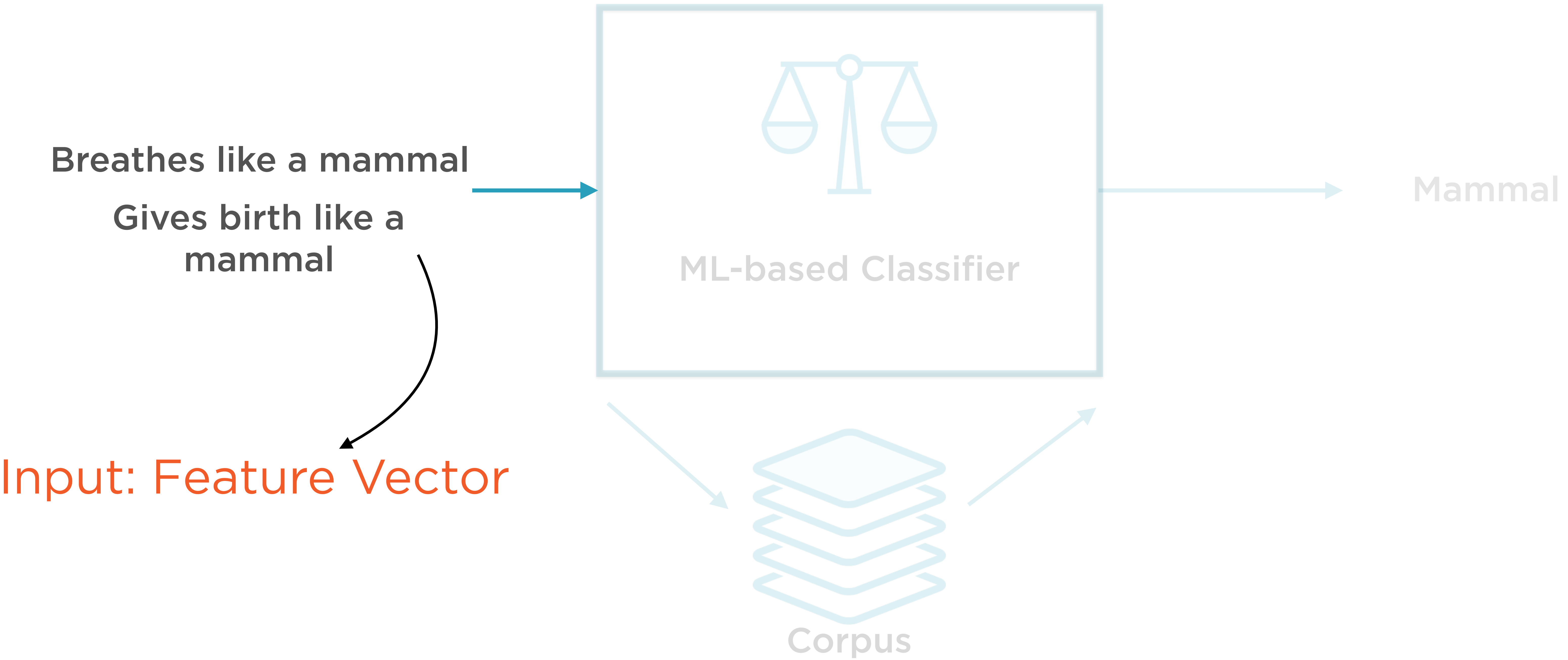
Look like fish, swim like fish, move with fish

Whales: Fish or Mammals?

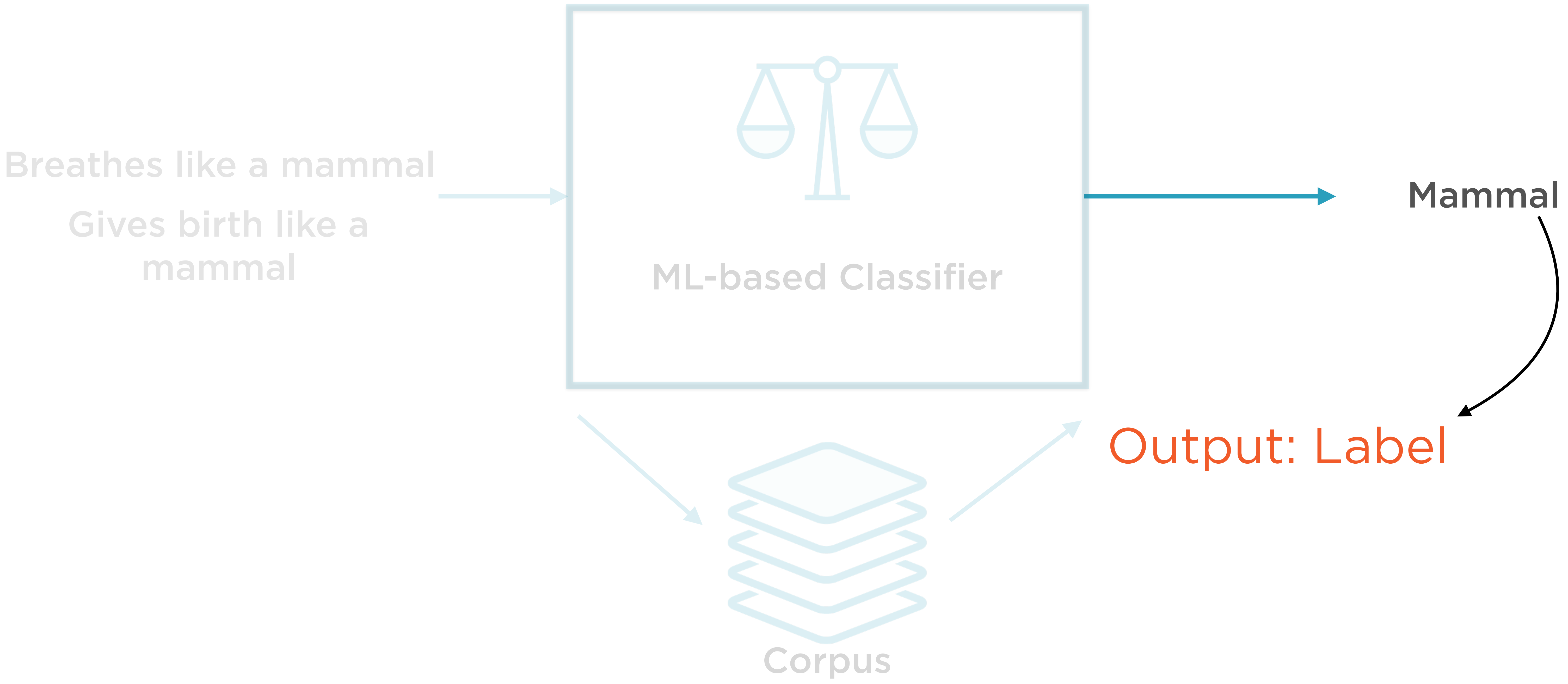
ML-based Binary Classifier



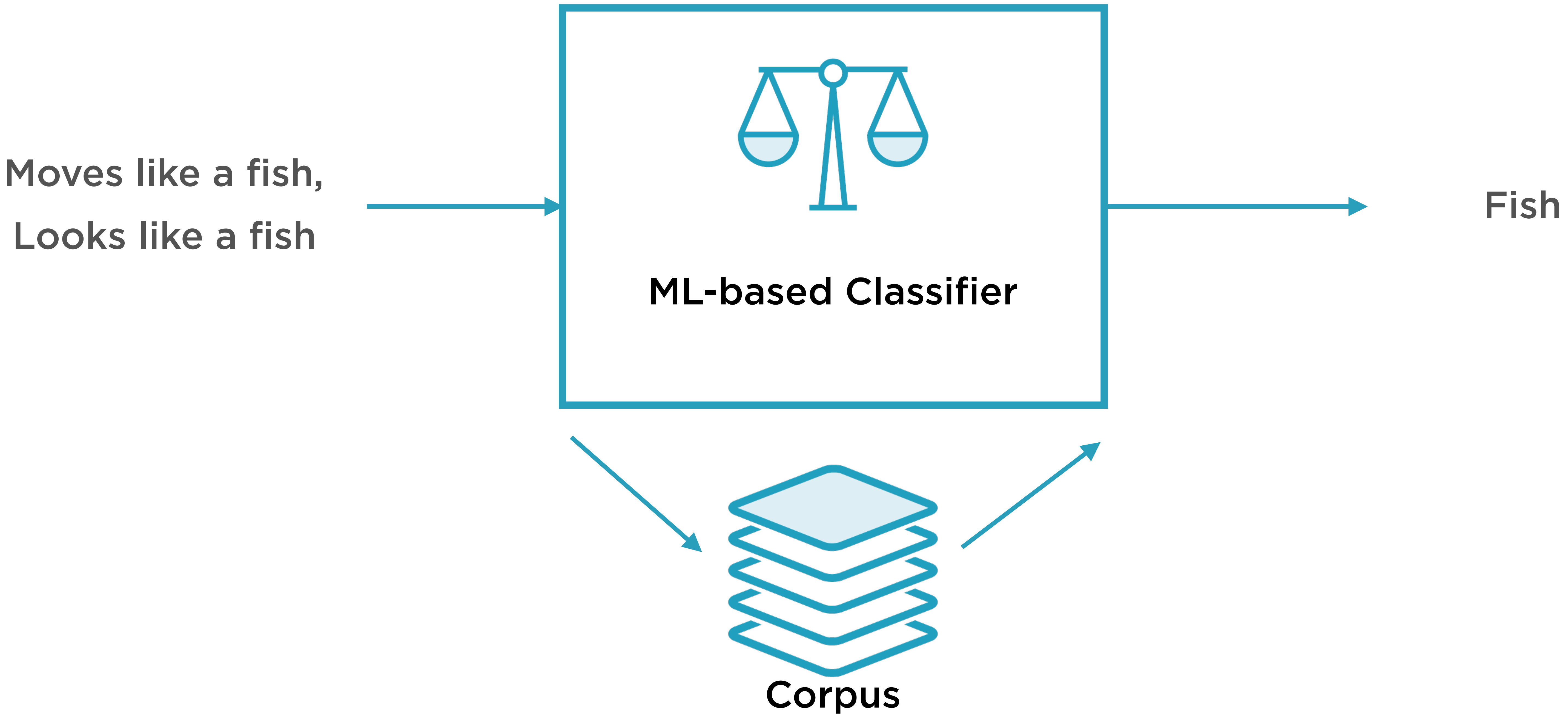
ML-based Binary Classifier



ML-based Binary Classifier



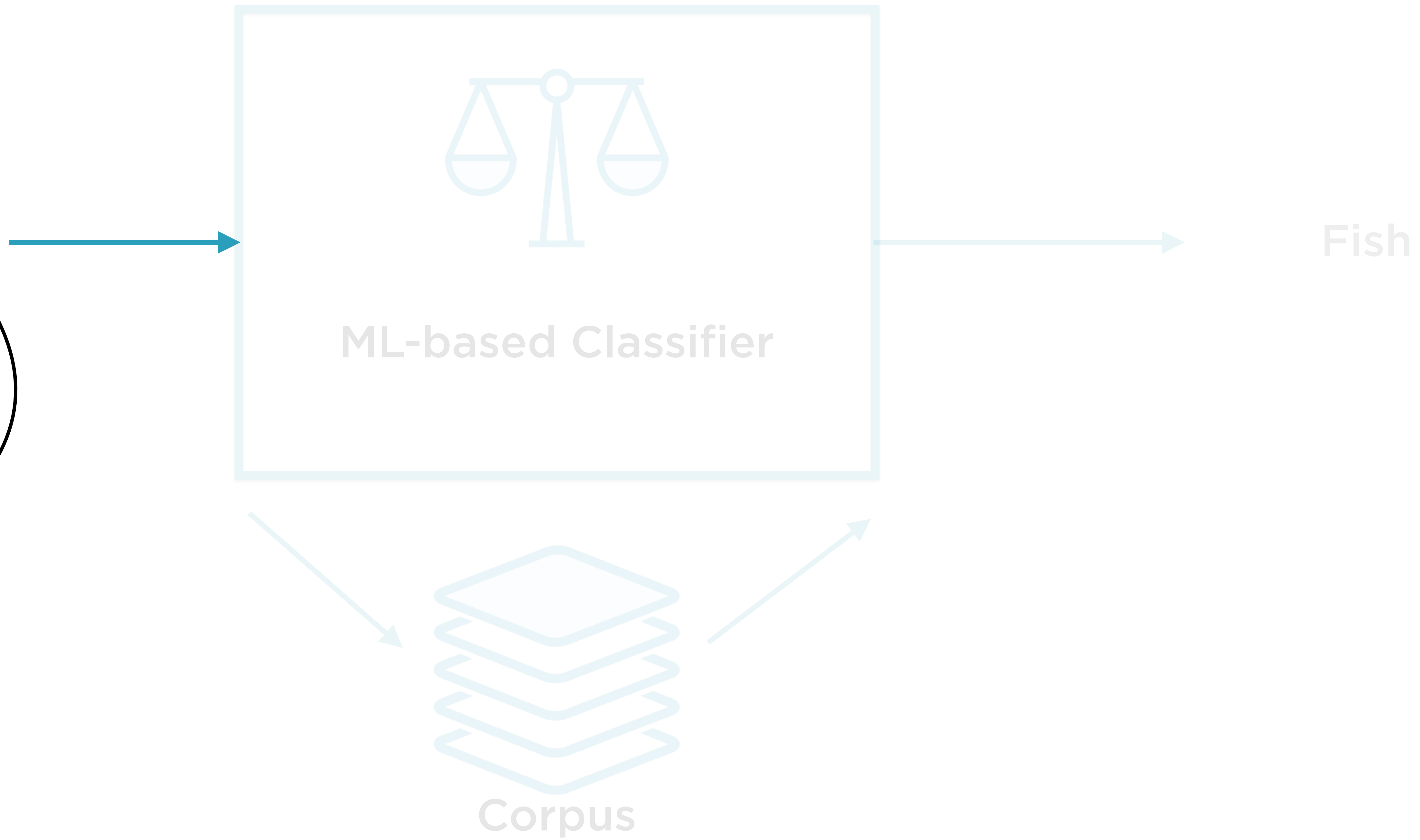
ML-based Binary Classifier



ML-based Binary Classifier

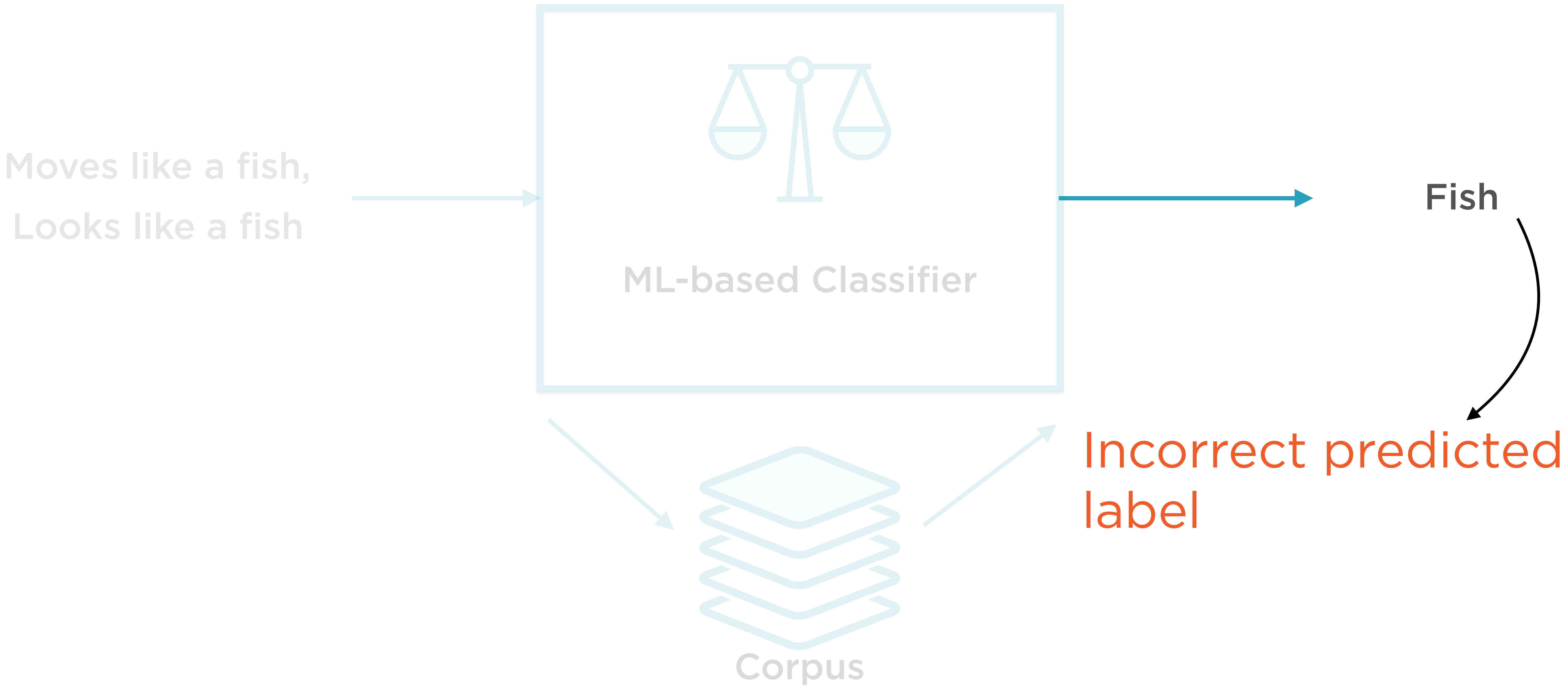
Moves like a fish,

Looks like a fish



Poor choice of features

ML-based Binary Classifier



To successfully apply ML algorithms to images, we need to capture the right features from images

Feature Detection

In the context of image processing, algorithms that detect the appropriate, most interesting, features from images.

Feature Detection

Identify “right” feature representations of images

Starting point of many computer vision algorithms

Repeatability an important factor

Whether the same feature will be detected in two or more images

Feature Detection

Abstractions such as style, texture

Content such as edges, corners

Feature Detection

Points of interest

- Corner points in edge detection
- Should be stable and repeatably identifiable

Regions of interest

- Blob detection in object tracking

Feature Detection

Edge detection

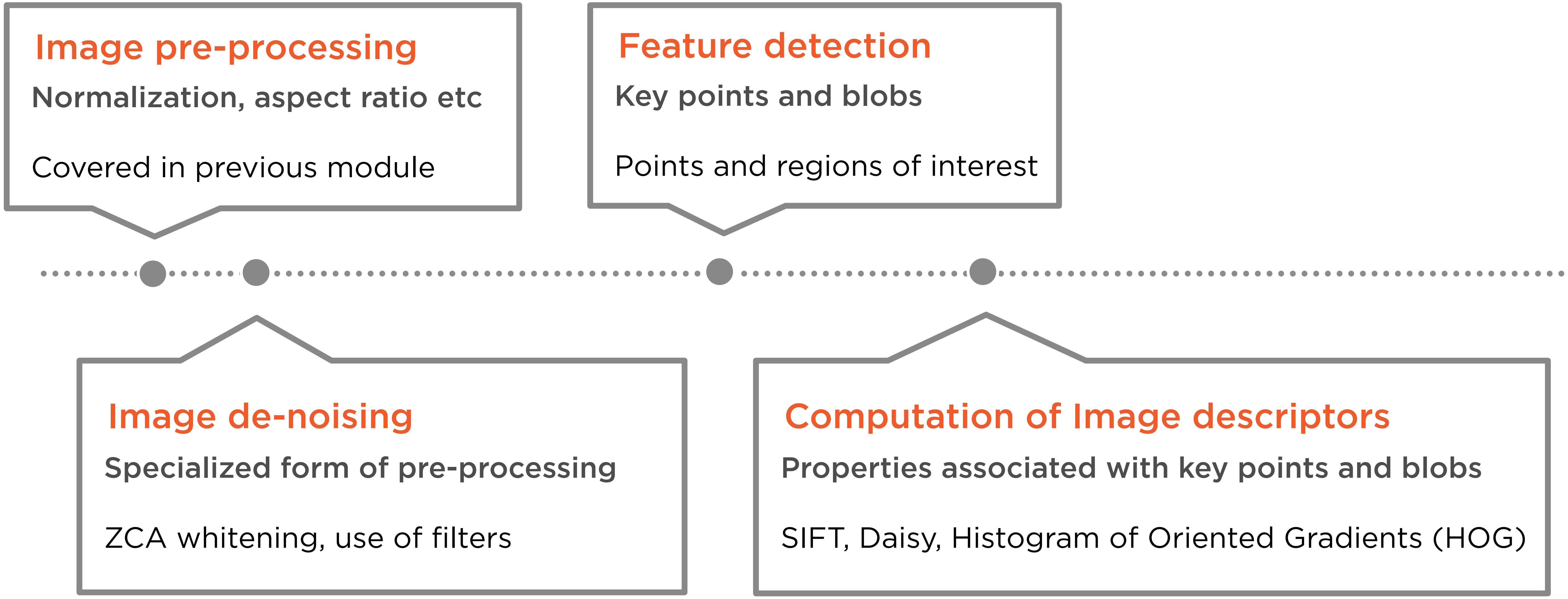
- Boundary between two image regions
- Can be of arbitrary shape

Ridge detection

- One dimensional curve which represents an axis of symmetry

Key Points and Descriptors

Using Images in Machine Learning



Feature Detection

Which points are most interesting?

Which regions are most interesting?

What is interesting about them?

Feature Detection

Detection of interest points

Which regions are most interesting?

What is interesting about them?

Feature Detection

Detection of interest points

Detection of blobs (areas) of interest

What is interesting about them?

Feature Detection

Detection of interest points

Detection of blobs (areas) of interest

Computation of image descriptors

Feature Detection

Detection of interest points

Detection of blobs (areas) of interest

Computation of image descriptors

Interest point detection

文A

This article needs additional citations for verification.

Learn more

Interest point detection is a recent terminology in computer vision that refers to the detection of interest points for subsequent processing. An interest point is a point in the image which in general can be characterized as follows:[1][2]

- It has a clear, preferably mathematically well-founded, definition,
- It has a well-defined *position* in image space,
- The local image structure around the interest point is rich in terms of local *information contents* (e.g.: significant 2D texture[3]), such that the use of interest points simplify further processing in the vision system,
- It is *stable* under local and global perturbations in the image domain as illumination/brightness variations, such that the interest points can be reliably computed with high degree of *repeatibility*.
- Optionally, the notion of interest point should include an attribute of *scale*, to make it possible to compute interest points from real-life images as well as under scale changes.

Historically, the notion of interest points goes back to the earlier notion of corner detection, where corner features were in early work detected with the primary goal of obtaining robust, stable and well-defined image features for object tracking and recognition of three-dimensional CAD-like objects from two-dimensional images. In practice, however, most corner detectors are sensitive not specifically to corners, but to local image regions which have a high degree of variation in all directions. The use of interest points also goes back to the notion of regions of interest, which have been used to signal the presence of objects, often formulated in terms of the output of a blob detection step. While blob detectors have not always been included within the class of interest point operators, there is no rigorous reason for excluding blob descriptors from this class. For the most common types of blob detectors (see the article on blob detection), each blob descriptor has a well-defined point, which may correspond to a local maximum, a local maximum in the operator response or a centre of gravity of a non-infinitesimal region. In all other respects, the blob descriptors also satisfy the criteria of an interest point defined above.

Key Points (a.k.a Points of Interest)

Points in the image that define what is interesting and must be captured in the feature representation of the image.

Properties of Key Points

Should be well-defined

Should not be affected by operations such as

- Rotation
- Translation
- Expansion
- Warping

Properties of Key Points

Interest point = Point with well-defined position that can be clearly defined

Types of interest points

- Corners: Intersection of two edges
- Intensity maxima/minima
- Line endings

Feature Detection

Detection of interest points

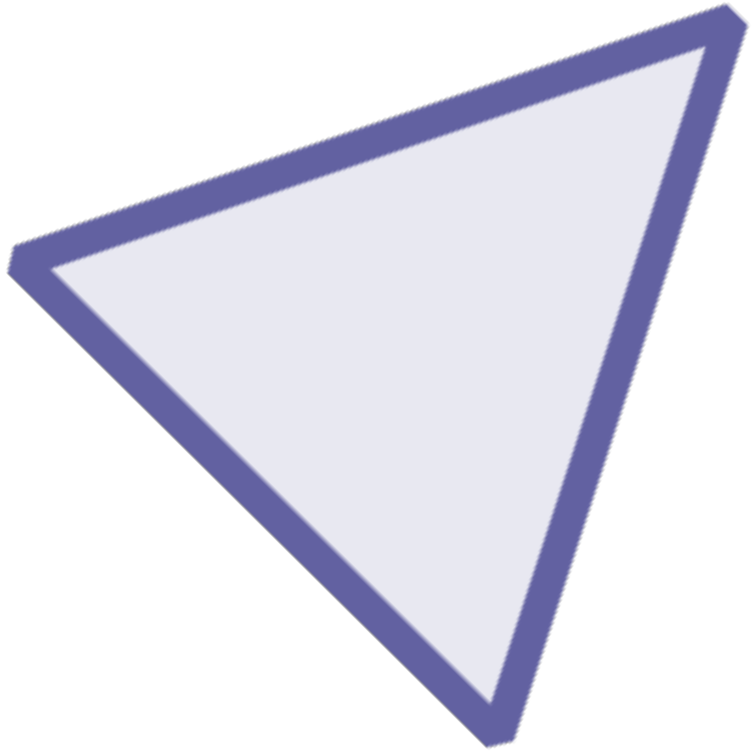
Detection of blobs (areas) of interest

Computation of image descriptors

Blobs (a.k.a Regions of Interest)

Interesting regions within an image within which points are similar and share properties that are different from surrounding points.

Blob Detection Applications

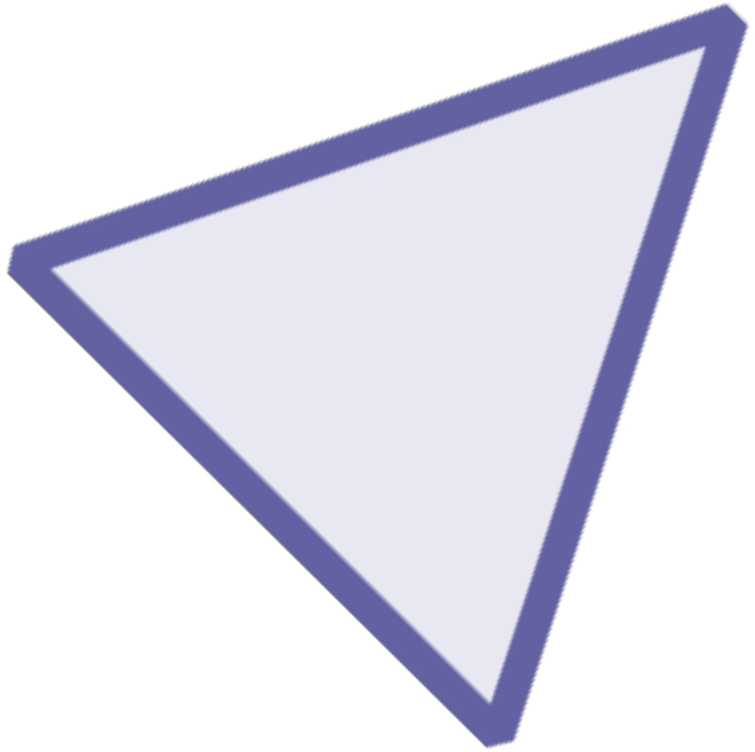


Complementary to points of interest

Capture additional information

- Object recognition
- Object motion tracking
- Texture analysis and detection
- Image segmentation

Blob Detection Algorithms



Two main families of algorithms

- Differential methods
- Local extrema methods

Blob detection

This article may be too technical for most readers to understand. Please help improve it to make it understandable to non-experts, without removing the technical details.

Learn more

In computer vision, **blob detection** methods are aimed at detecting regions in a digital image that differ in properties, such as brightness or color, compared to surrounding regions. Informally, a blob is a region of an image in which some properties are constant or approximately constant; all the points in a blob can be considered in some sense to be similar to each other. The most common method for blob detection is convolution.

Given some property of interest expressed as a function of position on the image, there are two main classes of blob detectors: (i) *differential methods*, which are based on derivatives of the function with respect to position, and (ii) *methods based on local extrema*, which are based on finding the local maxima and minima of the function. With the more recent terminology used in the field, these detectors can also be referred to as *interest point operators*, or alternatively interest region operators (see also interest point detection and corner detection).

There are several motivations for studying and developing blob detectors. One main reason is to provide complementary information about regions, which is not obtained from edge detectors or corner detectors. In early work in the area, blob detection was used to obtain regions of interest for further processing. These regions could signal the presence of objects or parts of objects in the image domain with application to object recognition and/or object tracking. In other domains, such as histogram analysis, blob descriptors can also be used for peak detection with application to segmentation. Another common use of blob descriptors is as main primitives for texture analysis and texture recognition. In more recent work, blob descriptors have found increasingly popular use as interest points for wide baseline stereo matching and to signal the presence of informative image features for appearance-based object recognition based on local image statistics. There is also the related notion of ridge detection to signal the presence of elongated objects.

Feature Detection

Detection of interest points

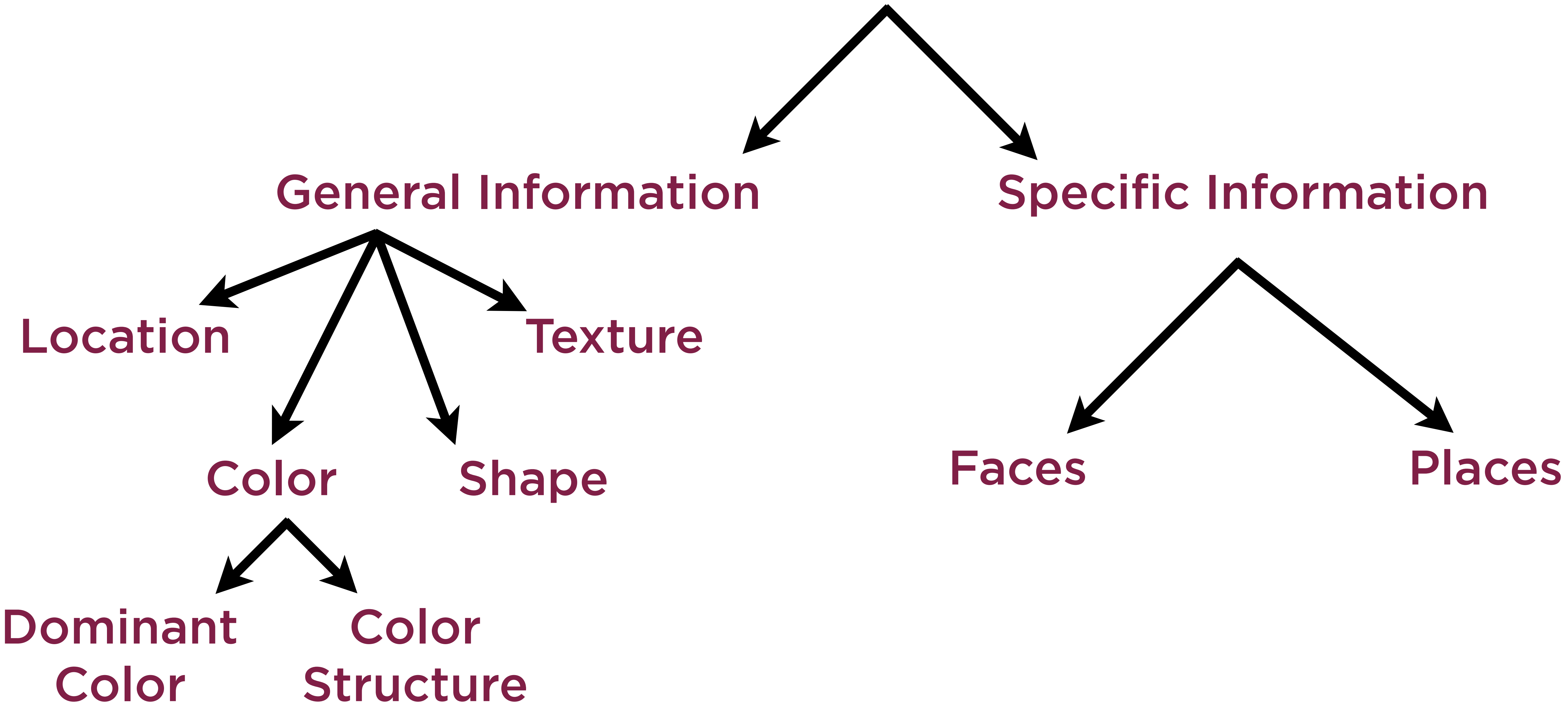
Detection of blobs (areas) of interest

Computation of image descriptors

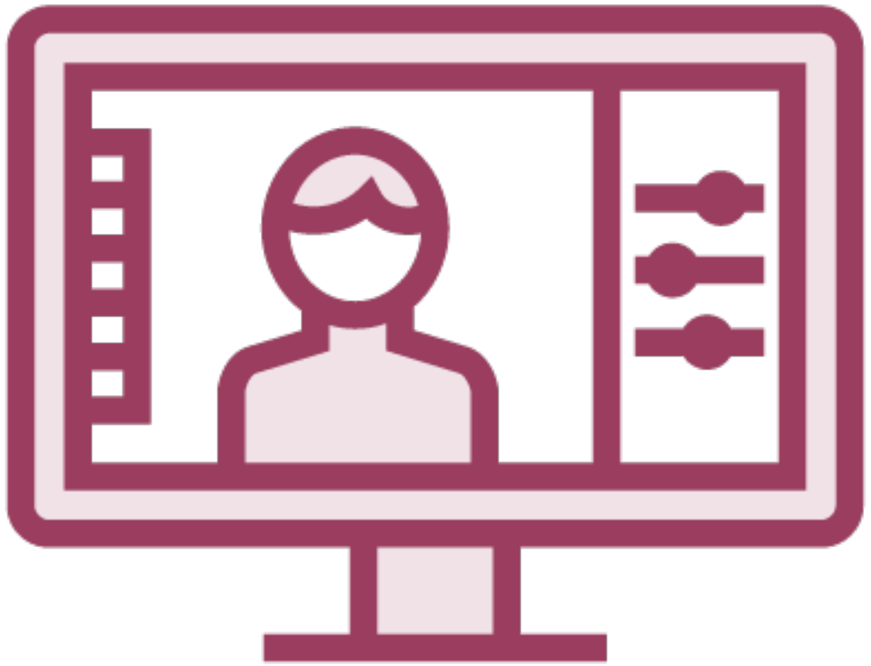
Image Descriptors

Descriptions of key features of images, such as shape, color, texture (and motion, in the case of videos)

Image Descriptors



Good Image Descriptors



Should be independent of position of associated key points

Should be robust to transformations

Should be scale independent

Image Descriptors

So, **here come descriptors**: they are the way to compare the keypoints. They summarize, *in vector format* (of constant length) some characteristics about the keypoints. For example, it could be their intensity in the direction of their most pronounced orientation. **It's assigning a numerical description to the area of the image the keypoint refers to.**

Some important things for descriptors are:

- they should be **independent of keypoint position**

  If the same keypoint is extracted at different positions (e.g. because of translation) the descriptor should be the same.
- they should be **robust against image transformations**

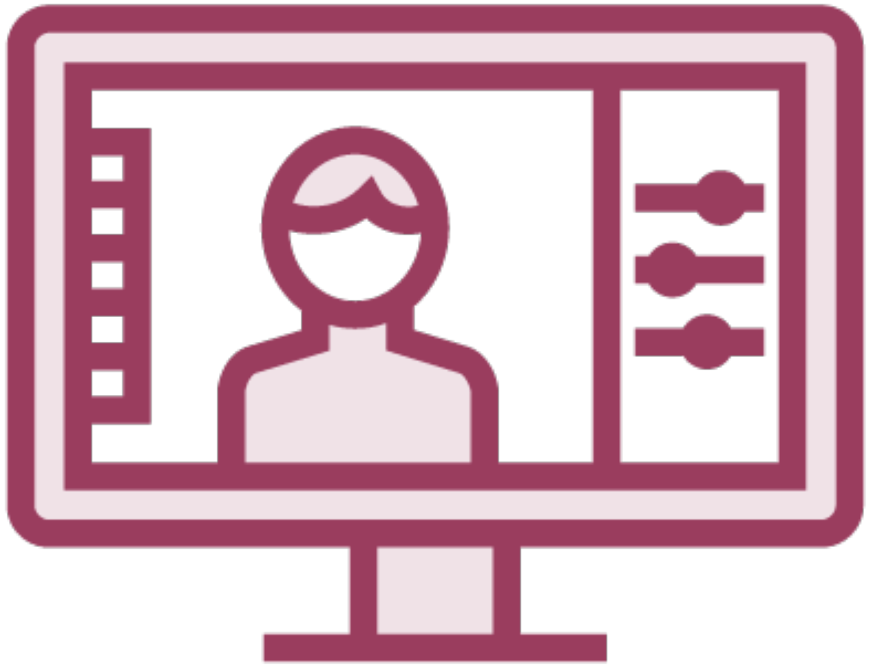
  Some examples are changes of contrast (e.g. image of the same place during a sunny and cloudy day) and changes of perspective (image of a building from center-right and center-left, we would still like to recognize it as a same building).

  Of course, no descriptor is completely robust against all transformations (nor against any single one if it is strong, e.g. big change in perspective).

  Different descriptors are designed to be robust against different transformations which is sometimes opposed to the speed it takes to calculate them.
- they should be **scale independent**

  The descriptors should take scale in to account. If the "prominent" part of the one keypoint is a vertical line of 10px (inside a circular area with radius of 8px), and the prominent part of another a vertical line of 5px (inside a circular area with radius of 4px) -- these keypoints should be assigned similar descriptors.

Image Descriptors for Feature Matching



Descriptors are vectors of numbers

Help compare key points across images

Can use distance measures to compare

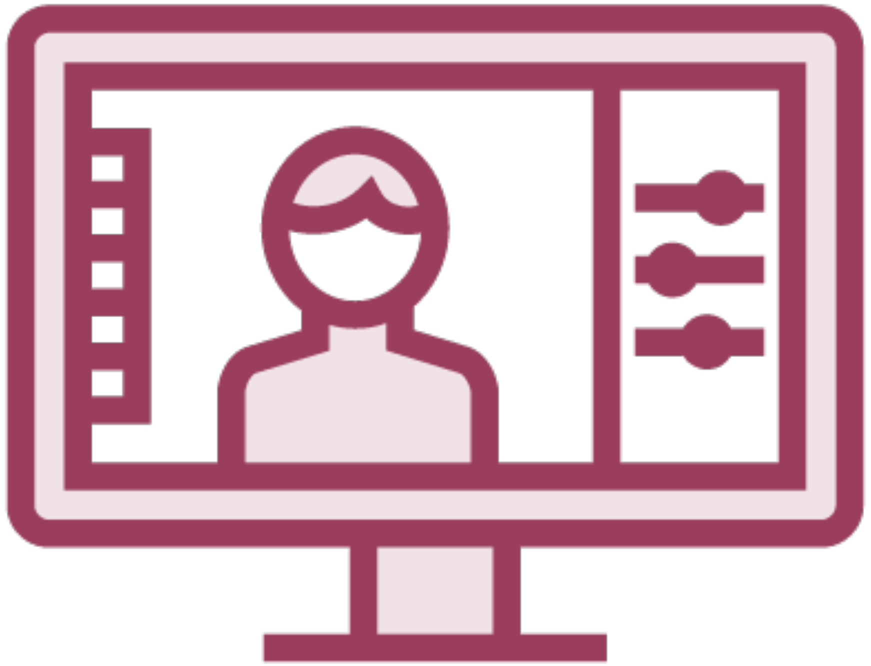
Key points whose descriptors have the smallest distances are matches

Demo

Apply key point preserving augmentations

SIFT, DAISY, HOG

Good Image Descriptors

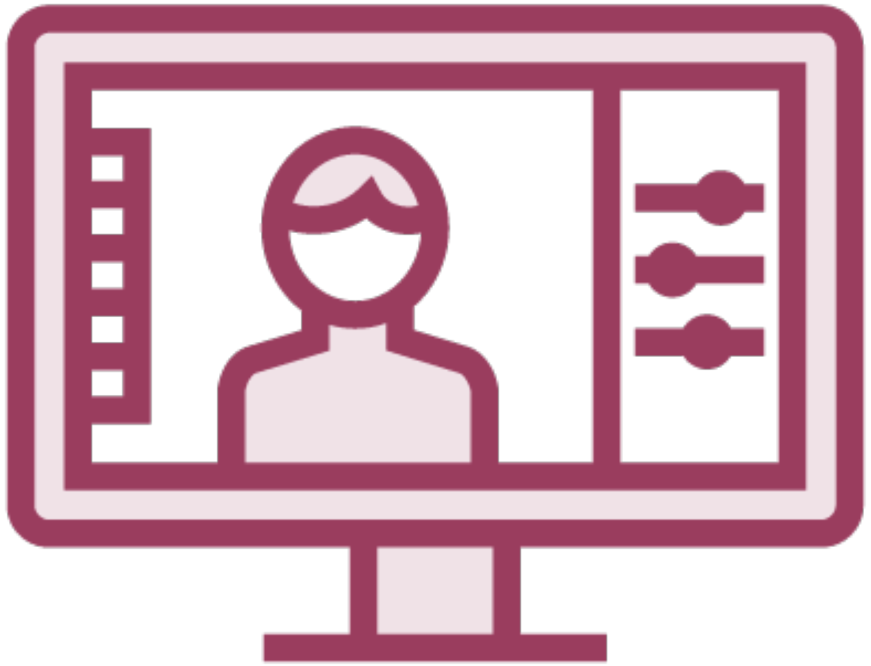


Should be independent of position of associated key points

Should be robust to transformations

Should be scale independent

Image Descriptors



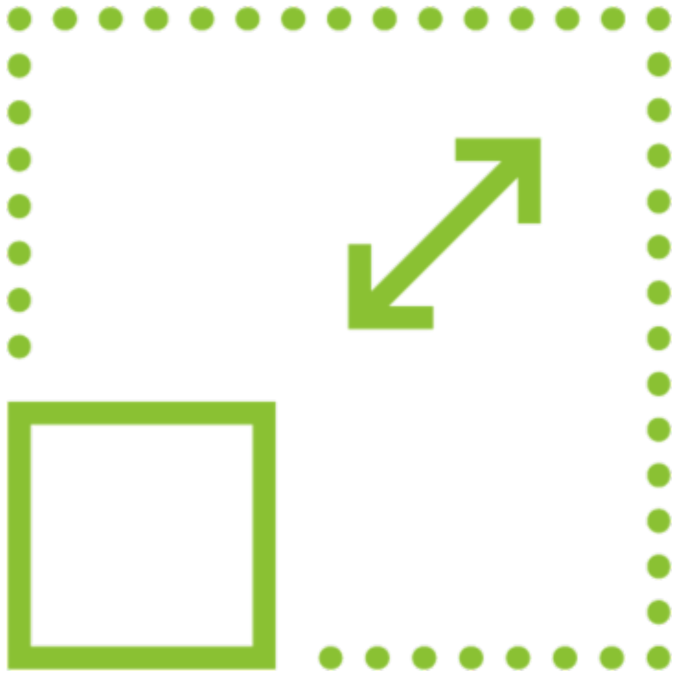
*S*cale *I*nvariant *F*eature *T*ransform (SIFT)

DAISY descriptors

Scale *I*nvariant *F*eature *T*ransform (SIFT)

Feature detection algorithm used to detect and describe features in images in a manner robust to translation, scaling and rotation

SIFT



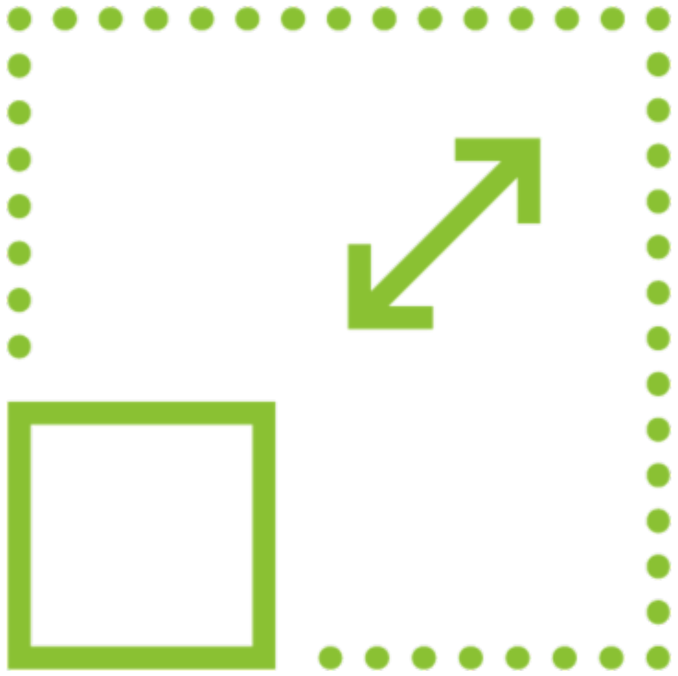
Start with corpus of reference images

Analyze and store descriptors

For new images, compare to corpus

Find matches with corpus database

SIFT



Invariant to scale and rotation

Robust against changes in illumination, noise and viewpoint

Highly distinctive

Robust to partial occlusion

SIFT

Scale Invariant Feature Detection

Convert images to large number of feature vectors

Feature Matching and Indexing

Efficiently store those feature vectors for fast key-based lookup

Cluster Identification

Given a fresh image, use Hough Transform to find all keys in corpus that this image matches

Model Verification

Minimize least square distance from original image and its feature vectors

Outlier Detection

Eliminate all feature vectors that are too far from original image

DAISY Descriptor

Feature selection algorithm, conceptually similar to SIFT, but faster and works with lower dimensionality feature vectors.

DAISY Algorithm

Also used for feature extraction

Dimensionality reduction

- Robust normalization
- Followed by PCA (Principal Components Analysis)

Histogram of Oriented Gradients

Feature descriptor used for object detection.

HOG

Image Normalization

Eliminate effects of illumination and shadows

Compute Gradients

Use simple first order gradients to find contour, silhouette and texture

Compute Histograms

Group cell histograms into larger, spatially connected blocks; aggregate into HOG by voting

Block Normalization

Divide each histogram block by L-1 norm or L-2 norm to normalize

Object Recognition

Use histogram blocks as feature vectors in your preferred ML algorithm

Block Normalization

Technique used to ensure that image histogram is not affected by lighting variations; relies on normalization of sub-matrices within histogram matrix.

Demo

Feature detection and extraction using SIFT

Demo

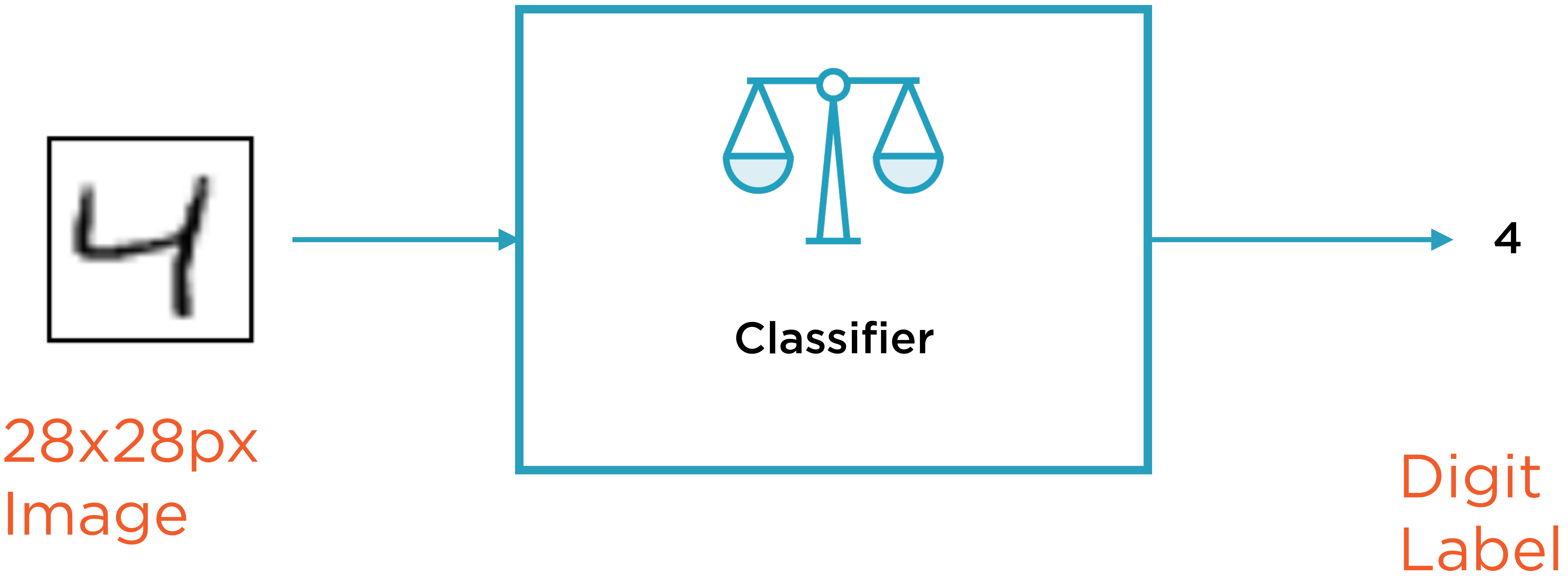
Feature detection and extraction using DAISY descriptors

Demo

Feature detection and extraction using the Histogram of Oriented Gradients (HOG) technique

Optical Character Recognition (OCR)

OCR Digit Recognition



Representing Images as Matrices

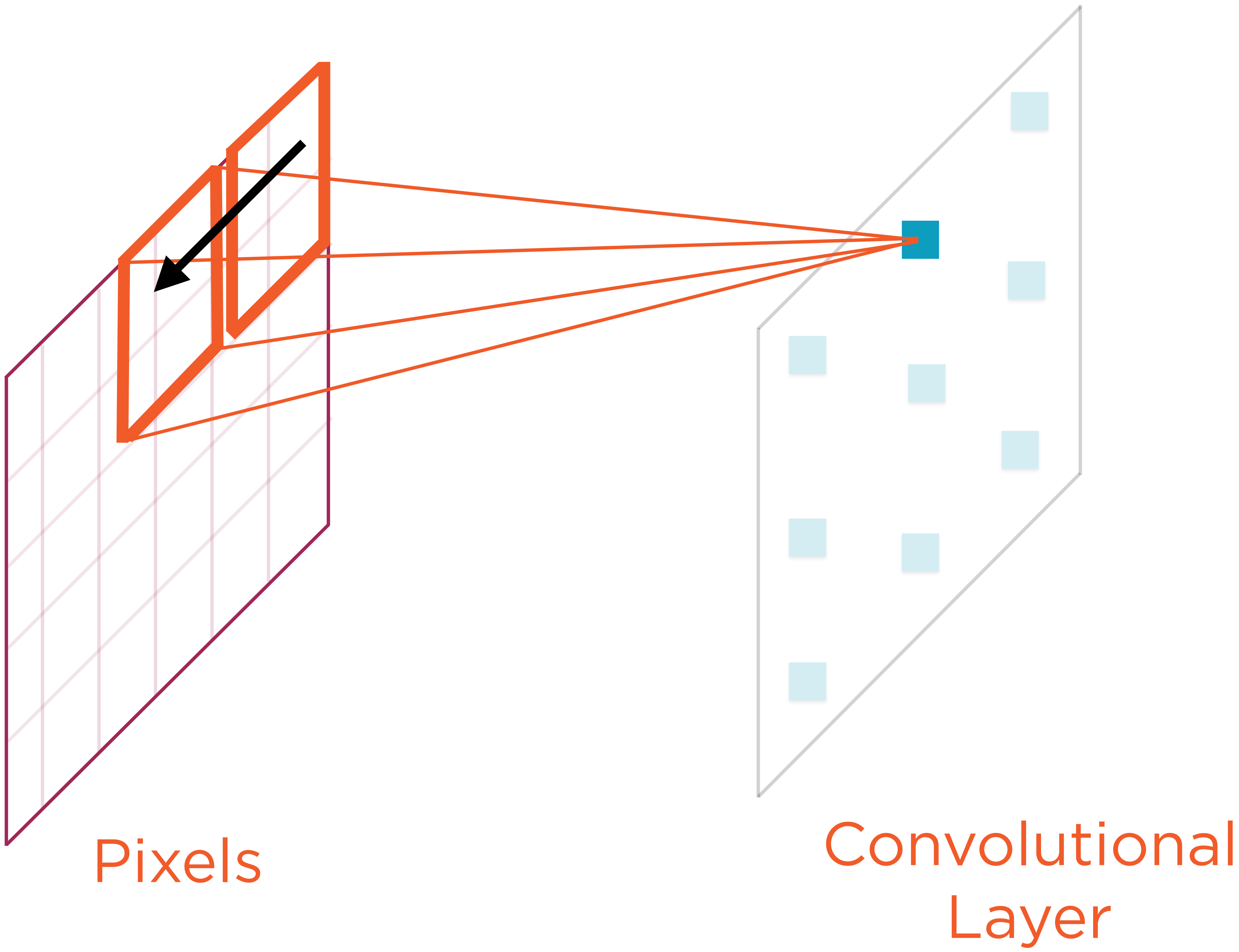
28

28

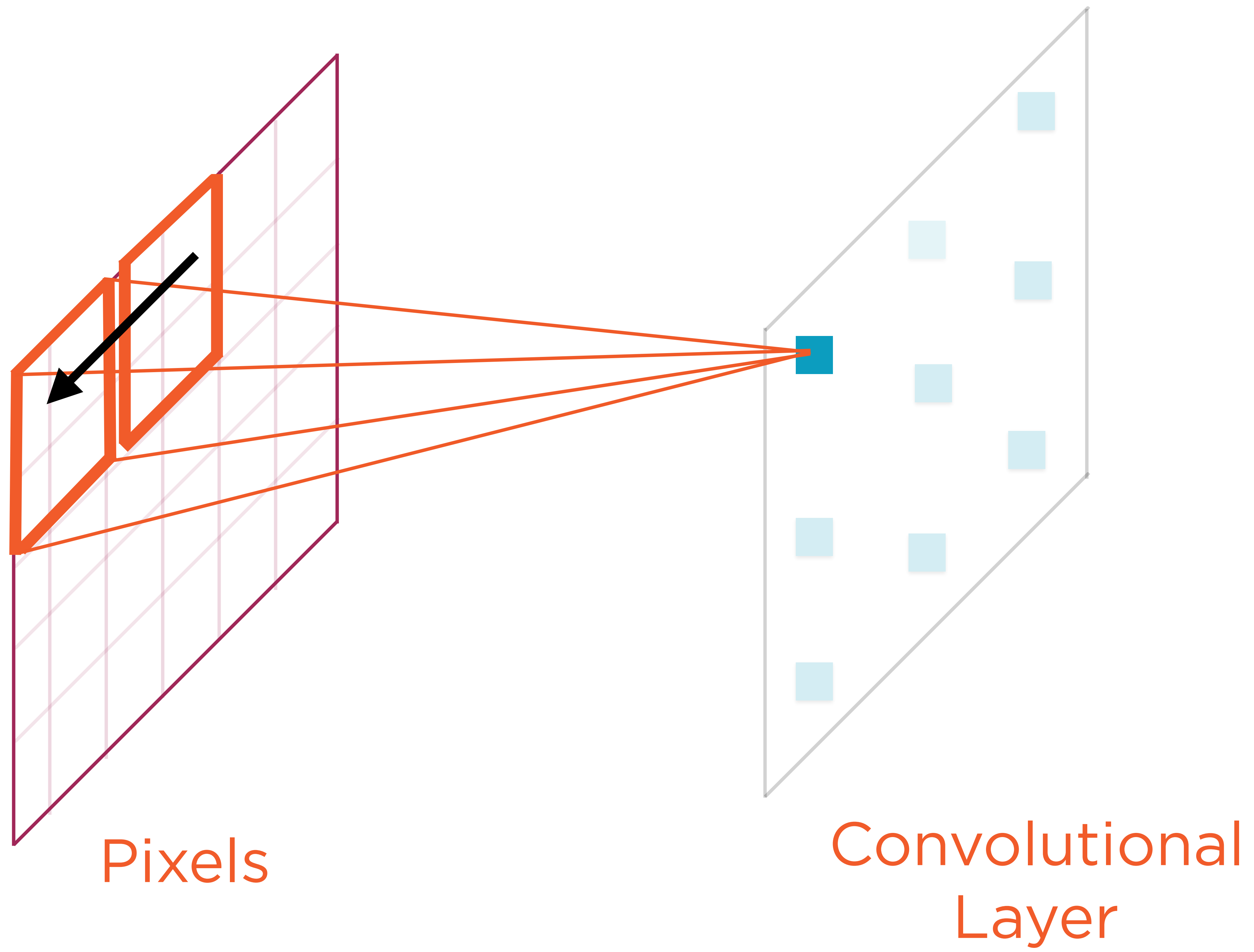
0	0	0	0	0	0
0.2	0.8	0	0.3	0.6	0
0.2	0.9	0	0.3	0.8	0
0.3	0.8	0.7	0.8	0.9	0
0	0	0	0.2	0.8	0
0	0	0	0.2	0.2	0

= 784 pixels

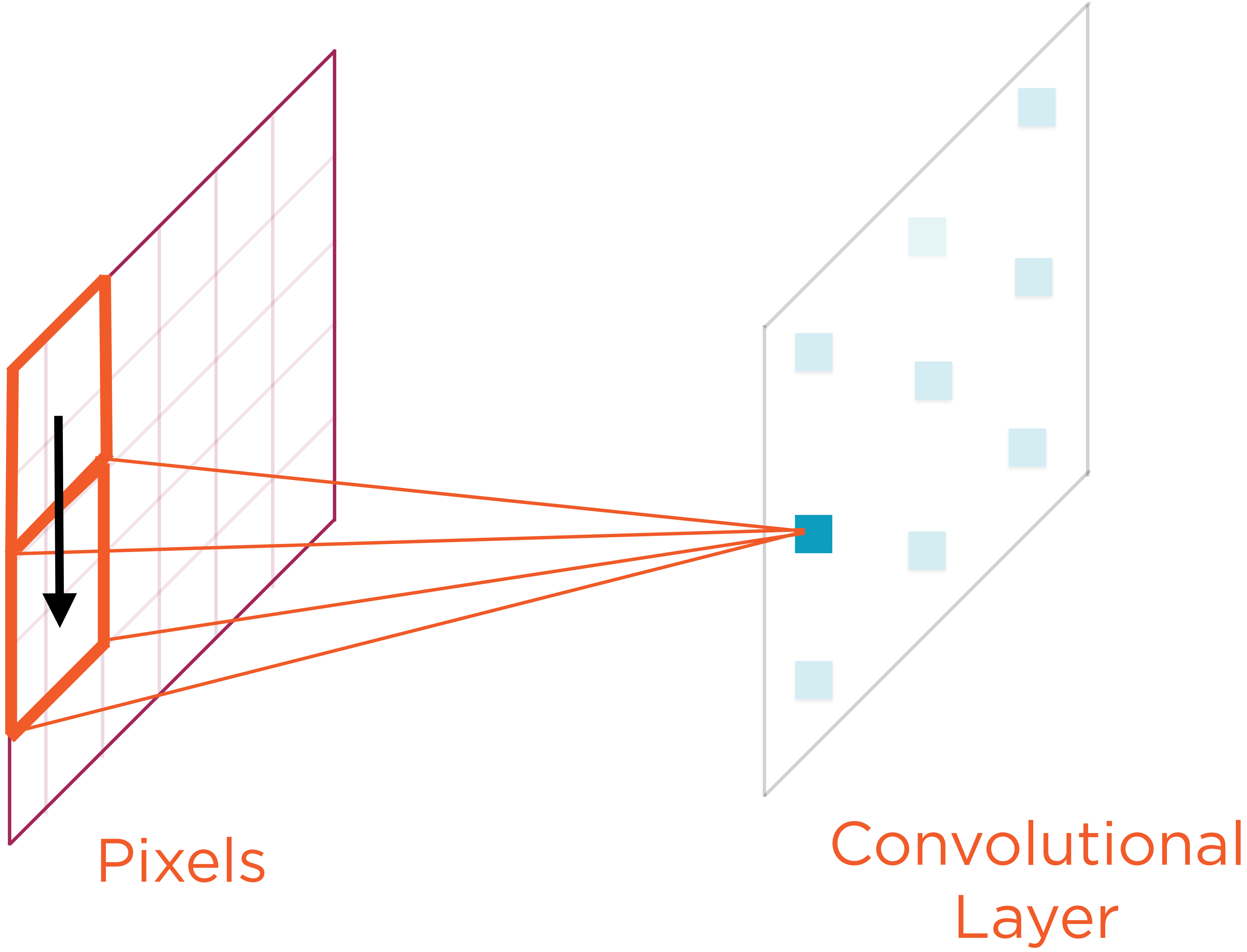
Convolution



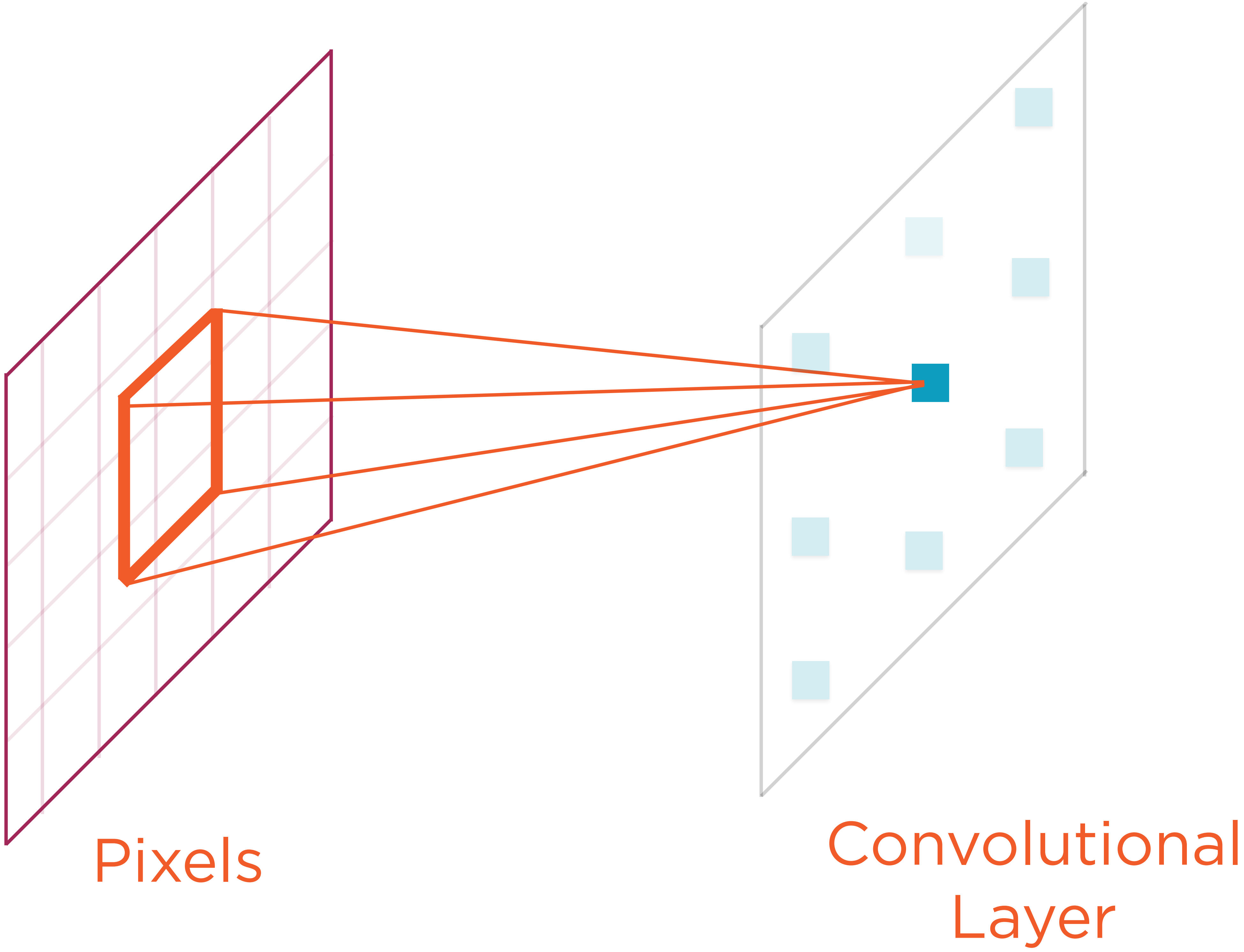
Convolution



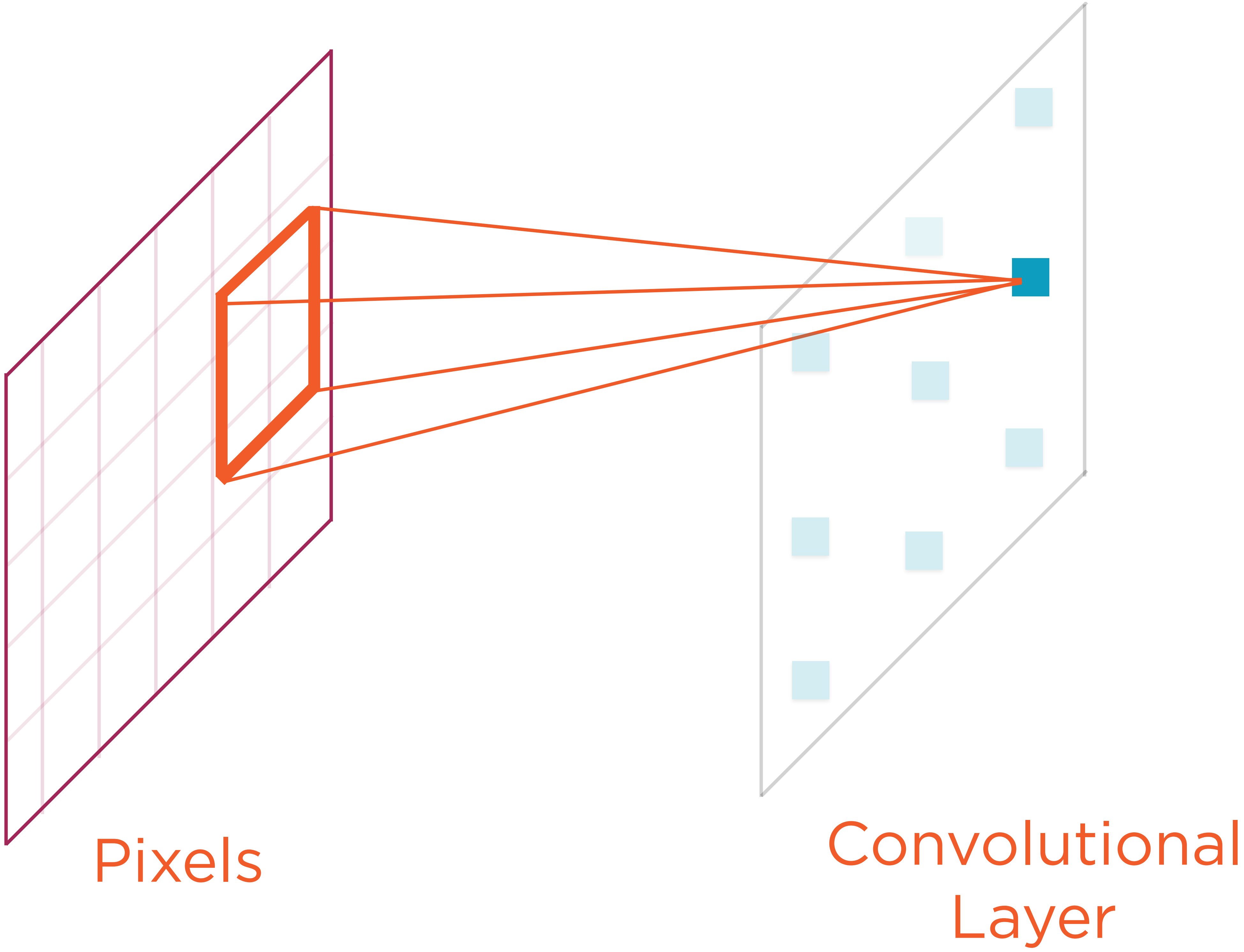
Convolution



Convolution

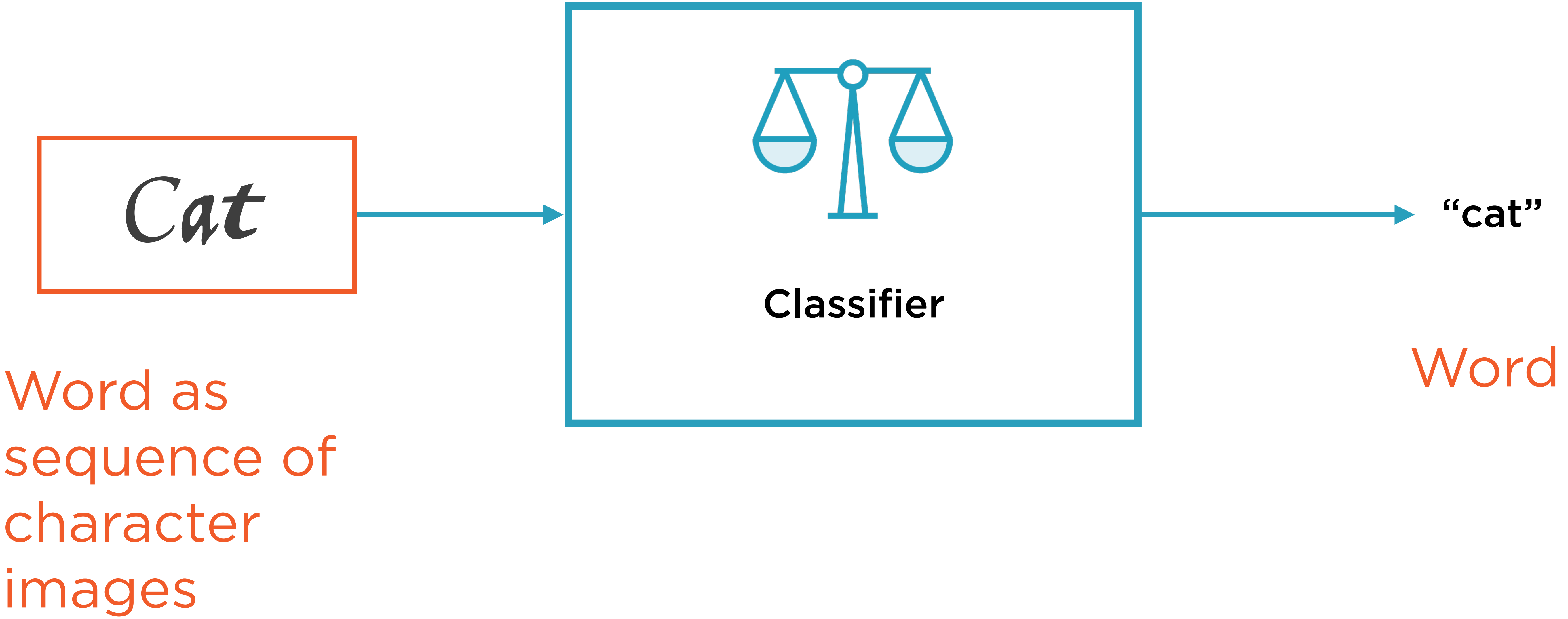


Convolution

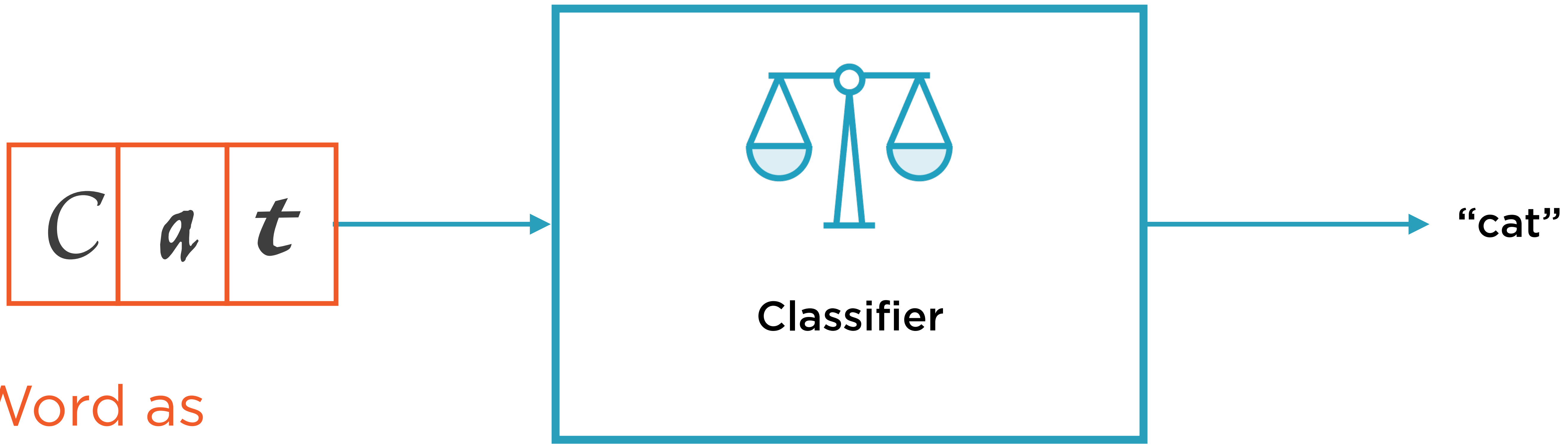


CNNs, when used in character recognition, identify patterns within a **single** image

OCR Word Recognition



OCR Word Recognition



Word as sequence of character images

Tesseract

Tesseract is an OCR engine with support for unicode and the ability to recognize more than 100 languages out of the box. It can be trained to recognize other languages.

Tesseract

Developed at Google, it is used for text detection on mobile devices, in video and Gmail image spam detection.

Python-tesseract

Python wrapper for Google's Tesseract OCR engine.

Demo

Optical character recognition using Tesseract

Summary

Importance of feature detection

Scale Invariant Feature Transform (SIFT) and DAISY

Histogram of Oriented Gradients (HOG)

Optical Character Recognition (OCR)