# Face Detection With OpenCV

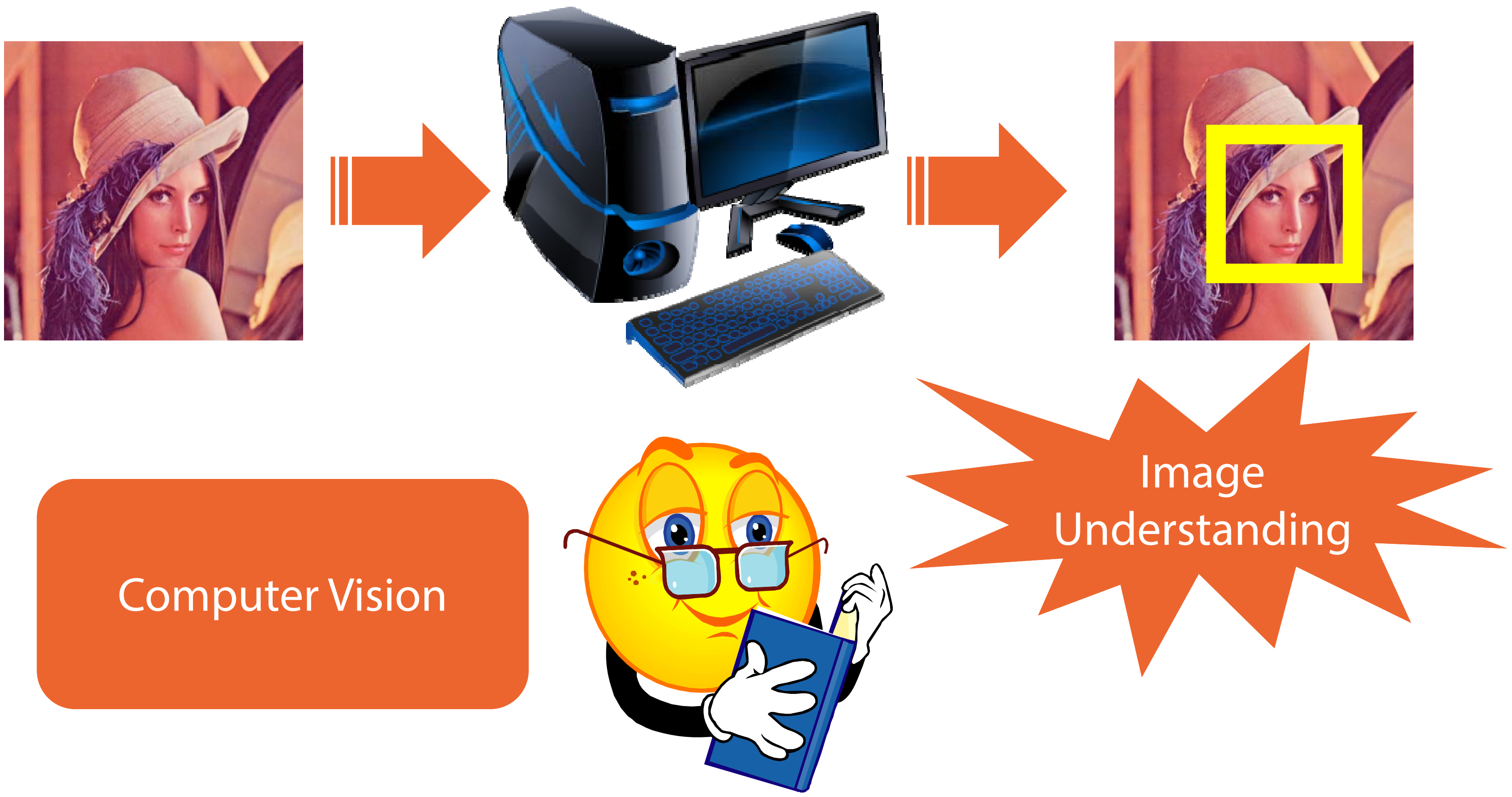
Giovanni Dicanio
giovanni.dicanio@gmail.com

hardcore dev and IT training

## Topics

- **Face detection**
  - OpenCV
- **OpenCV/Cinder integration**

## Face Detection



## Complexity

# OpenCV

**Open** Source
**C**omputer **V**ision
Library

OpenCV

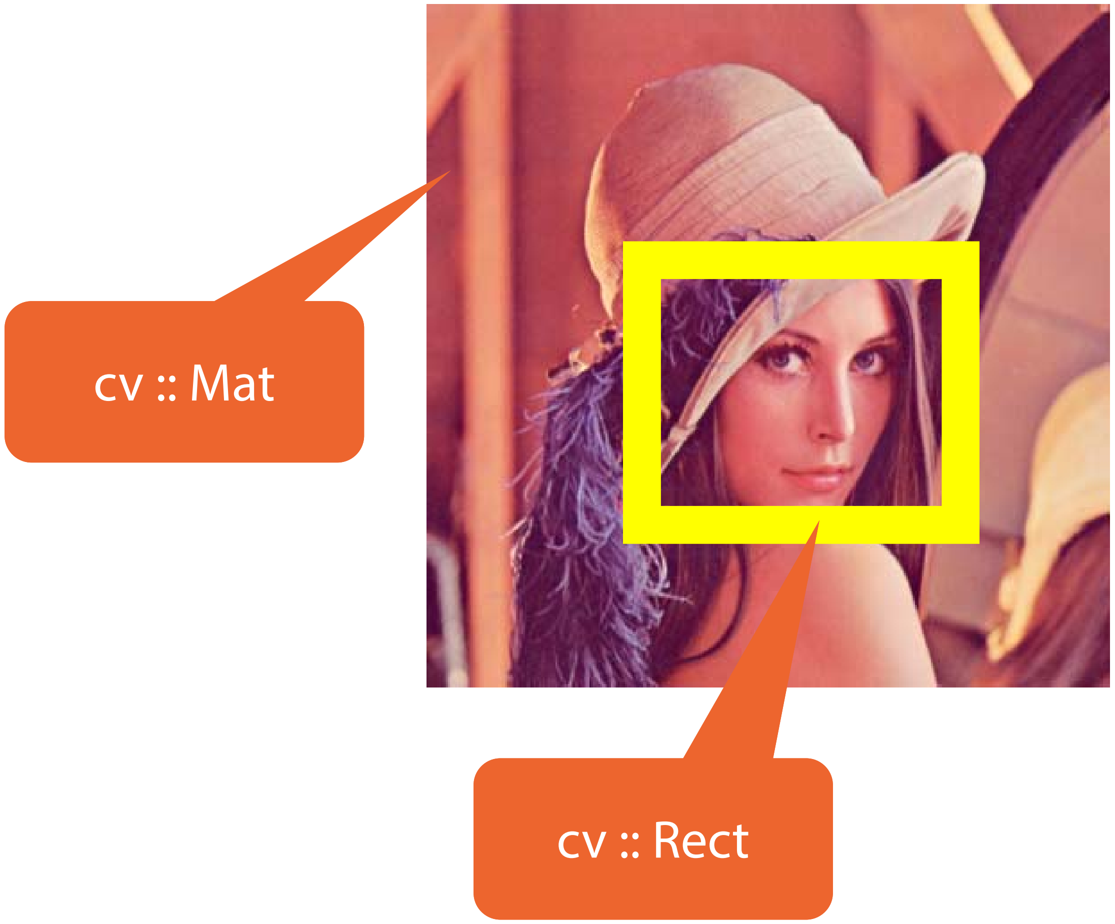
## What Does OpenCV Offer?

- Detecting faces
- Identifying objects
- Finding similar images from an image database
- …More than 2,500 optimized algorithms!

opencv.org

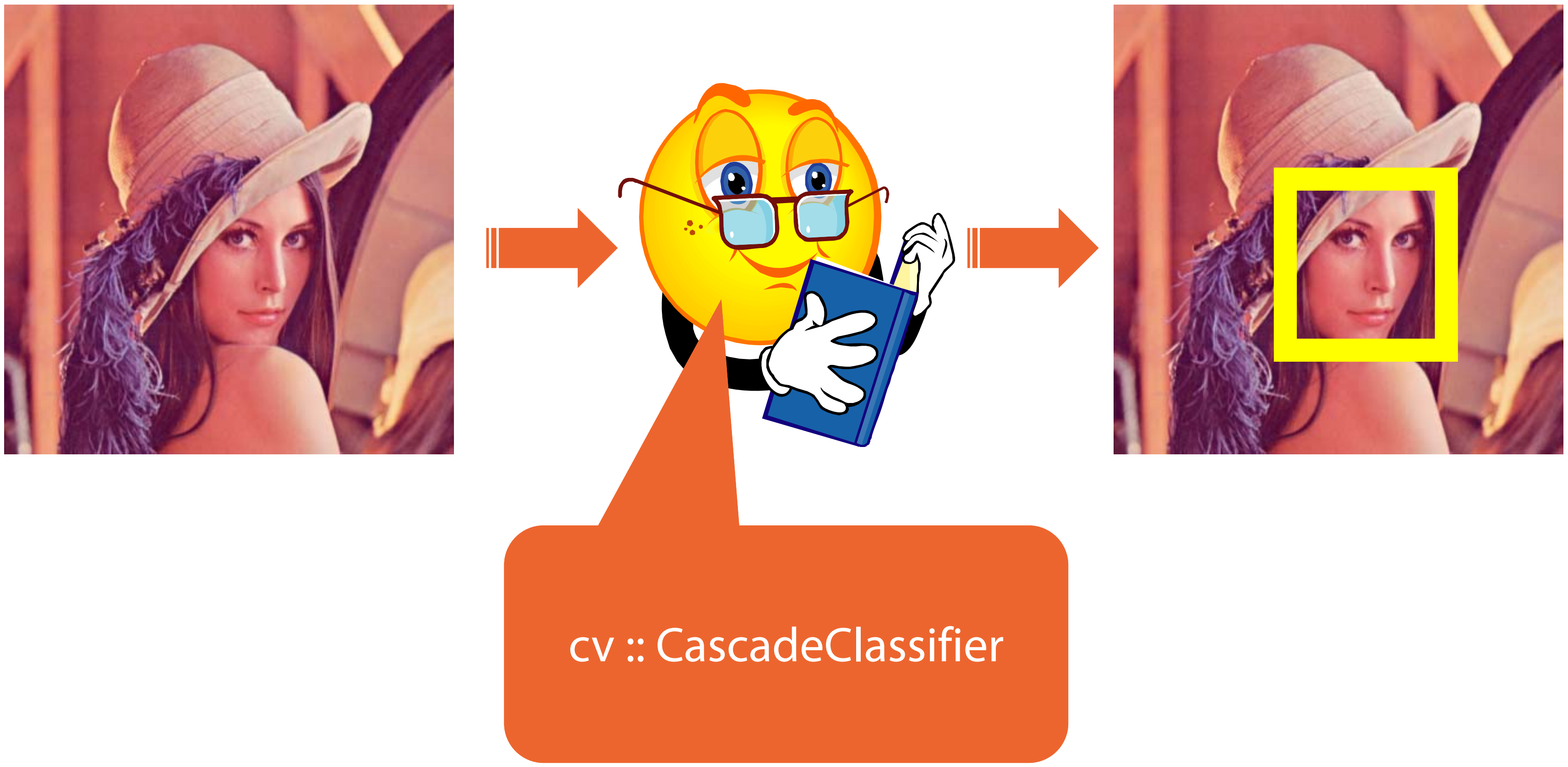
OpenCV

## Software Components for Face Detection

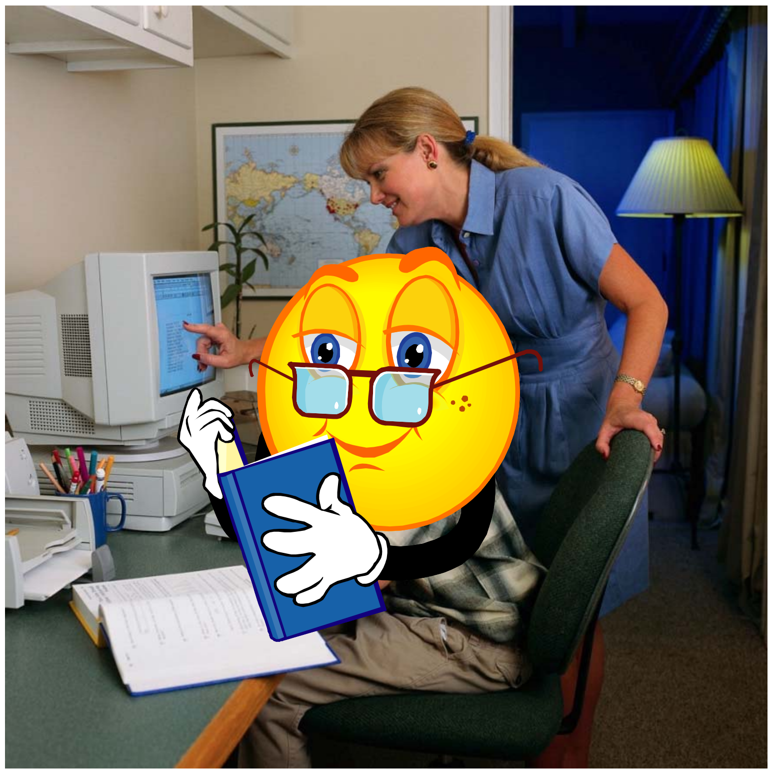


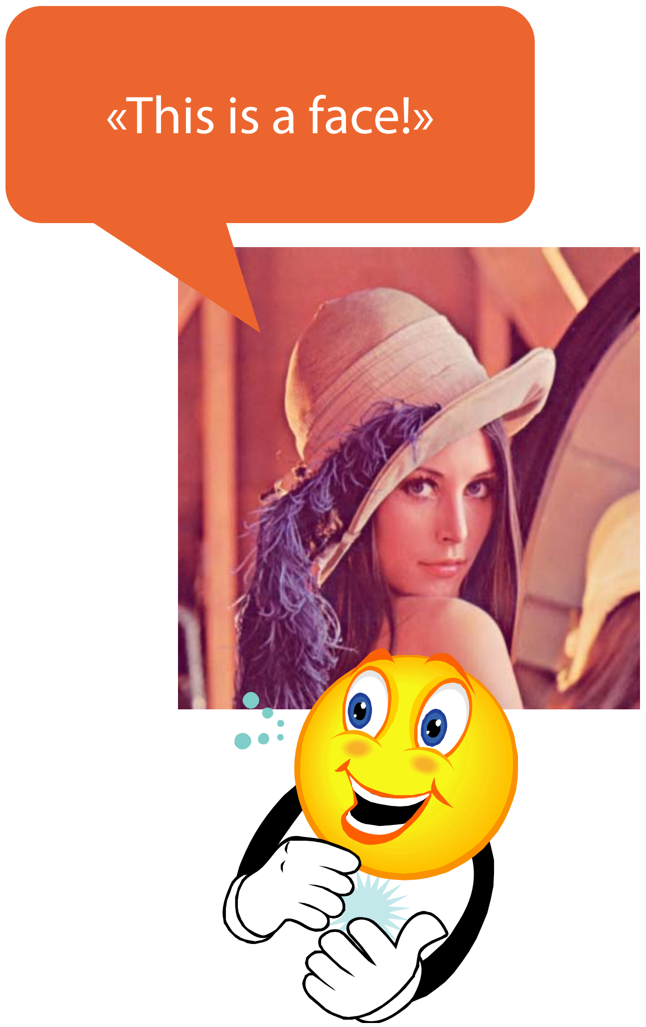
OpenCV

## The cv::CascadeClassifier Class

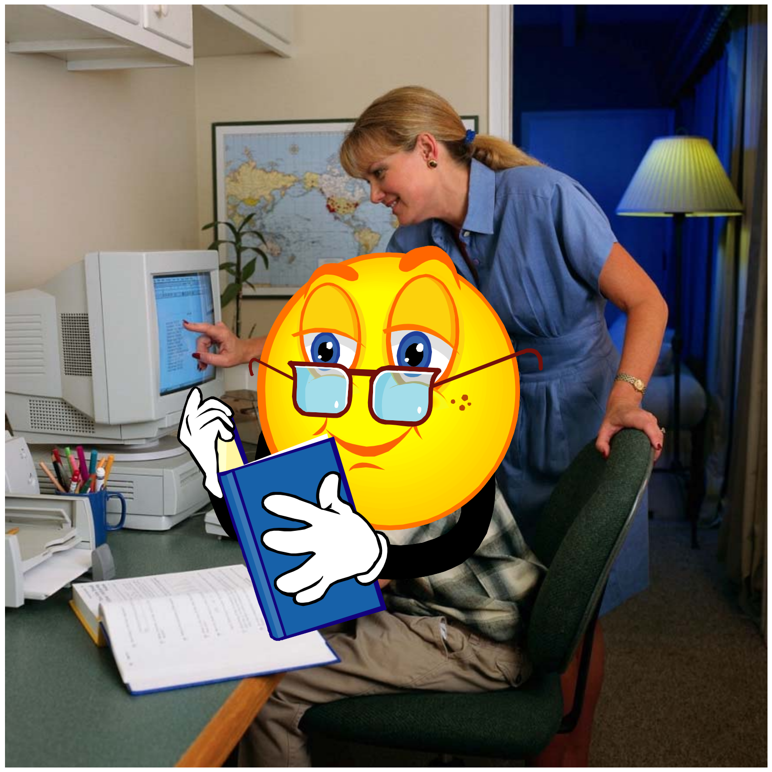


## Training the Classifier

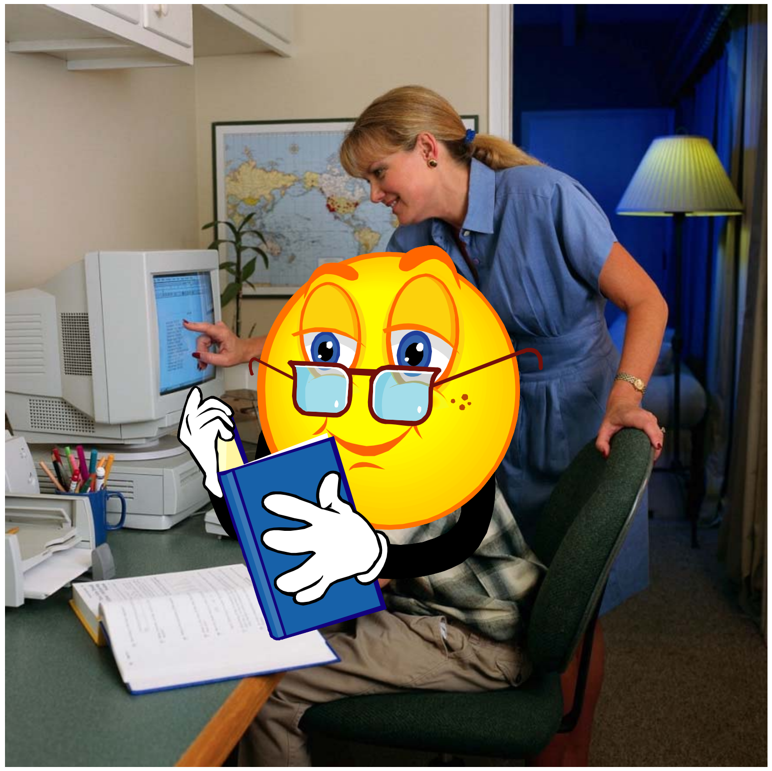




## Training the Classifier



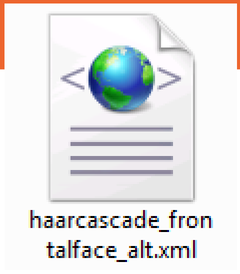
## Training the Classifier



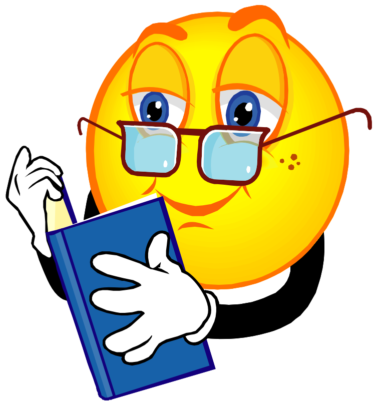
## Training the Classifier

http://bit.ly/1nV1aaj

CascadeClassifier :: **load** ()



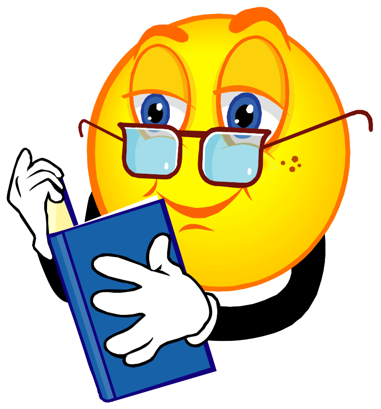
## Using the Classifier



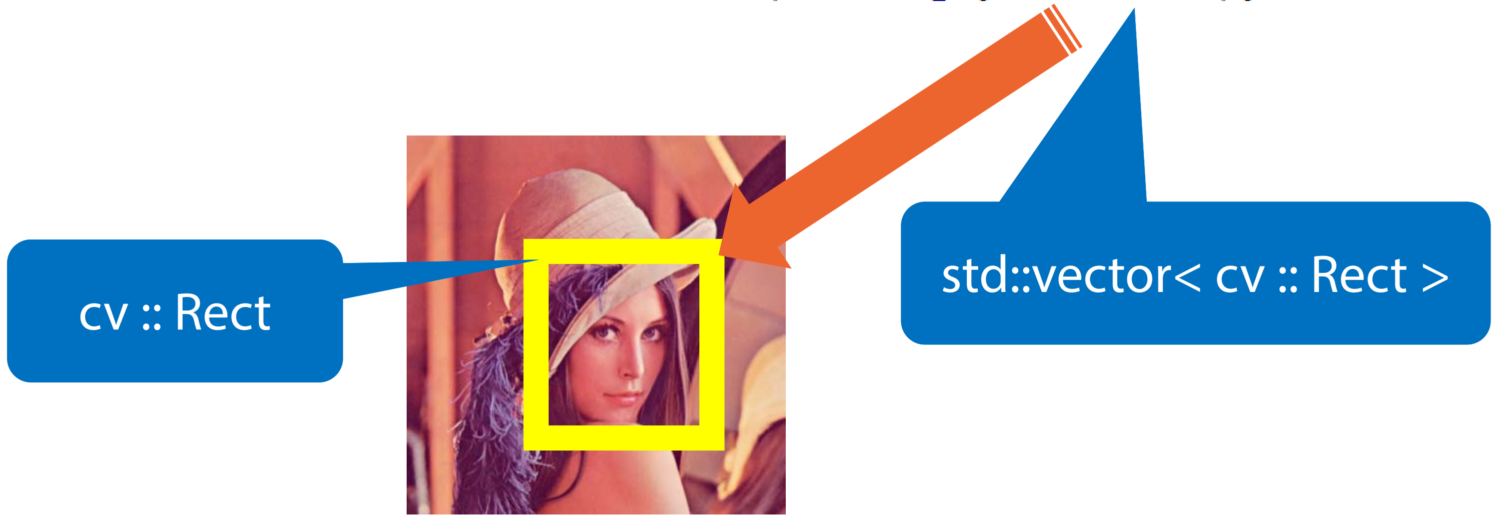
```
mFaceClassifier.detectMultiScale( cvImage, cvFaces );
```

cv :: Mat

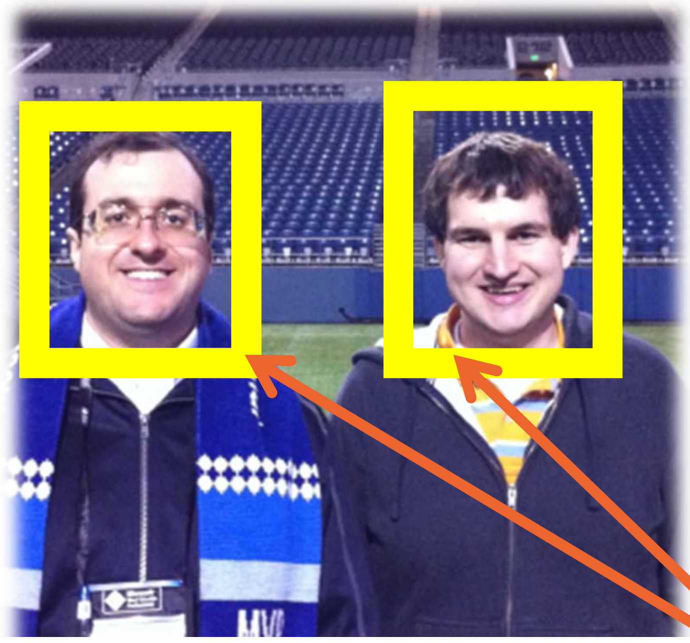
## Using the Classifier



```
mFaceClassifier.detectMultiScale( cvImage, cvFaces );
```



## Why std::vector?

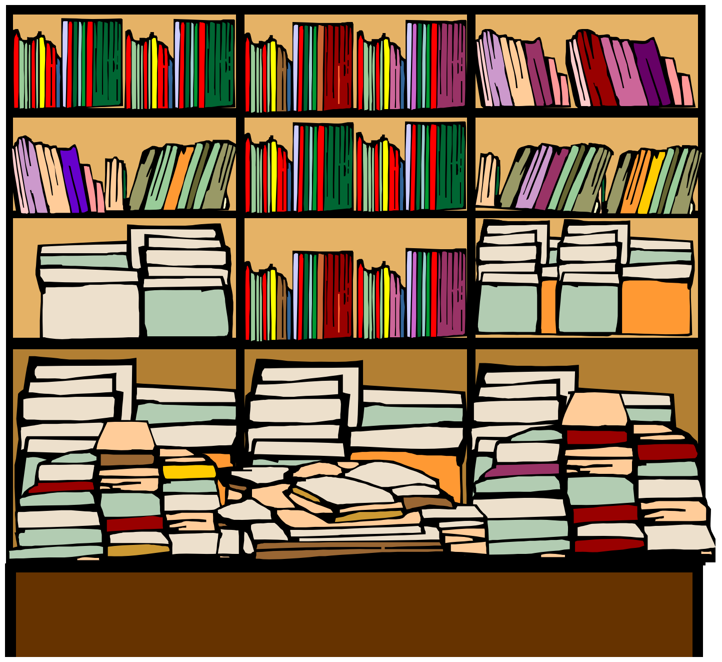


```
mFaceClassifier.detectMultiScale( cvImage, cvFaces );
```

std::vector< cv :: Rect >

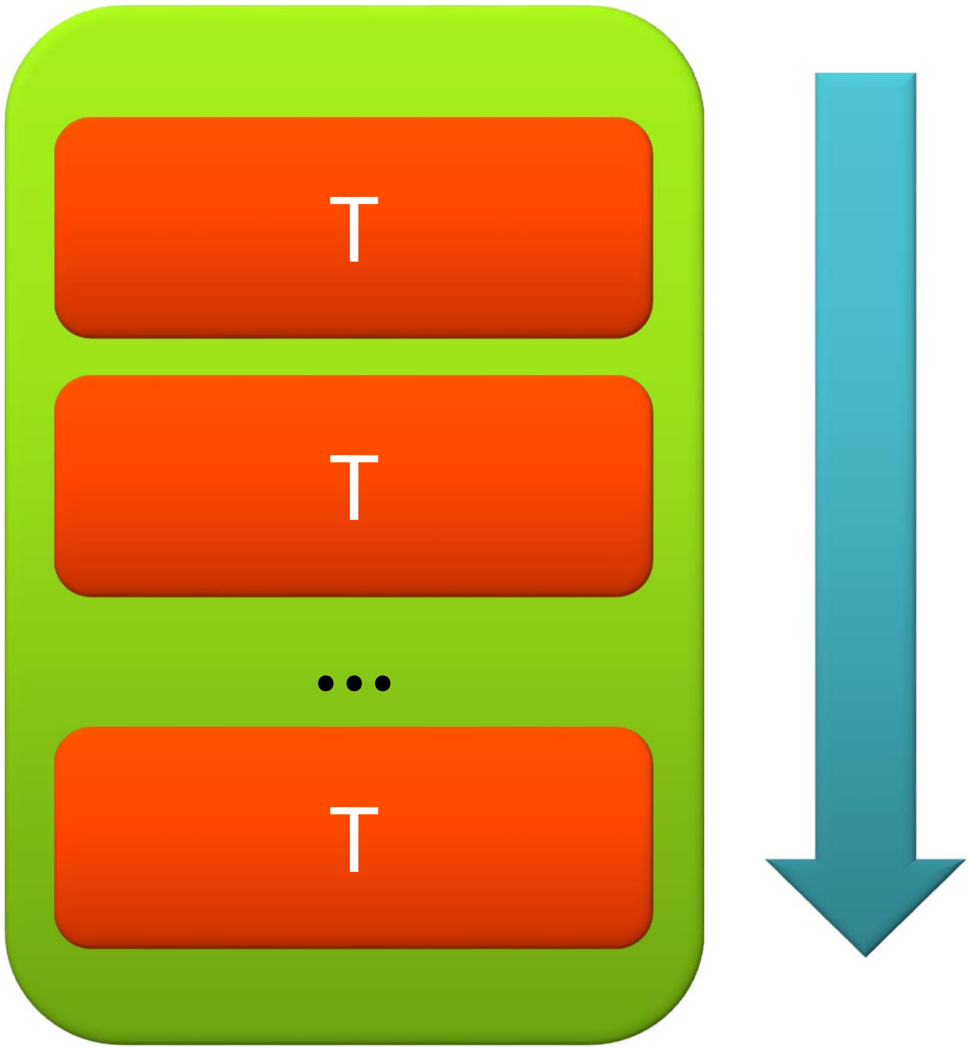
## The std::vector Container

STL

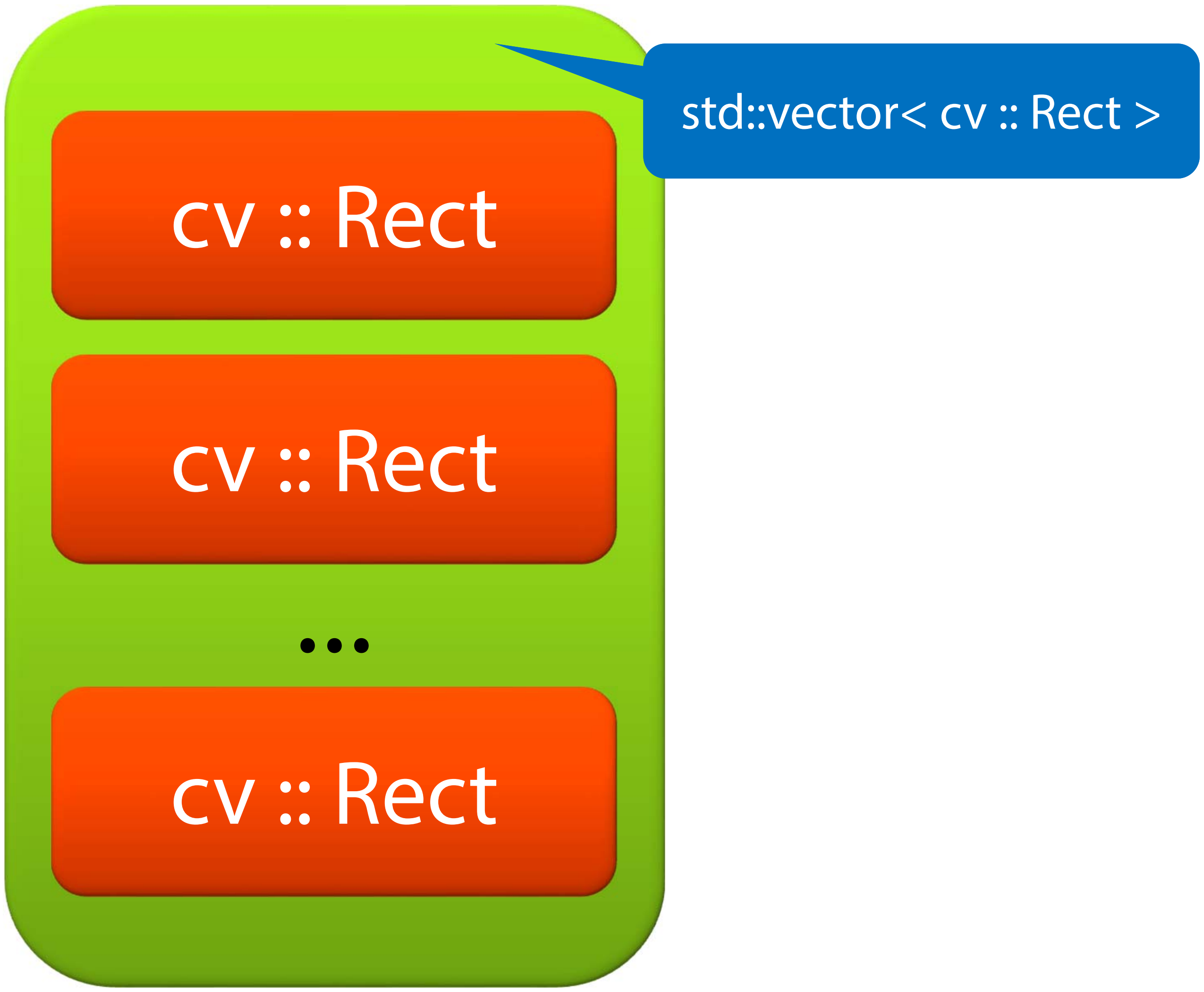


## The std::vector Container

The elements are stored **contiguously**.



# Vector of OpenCV Rectangles



## OpenCV Face Detection Recap

- **cv::Mat storing an image**
- **cv::CascadeClassifier object**
  - Load a classifier XML file for face detection
- **Call detectMultiScale()**
  - Result: face «rectangle(s)» passed in std::vector<cv::Rect>

## Exchanging Data Between Cinder and OpenCV

ci :: Surface

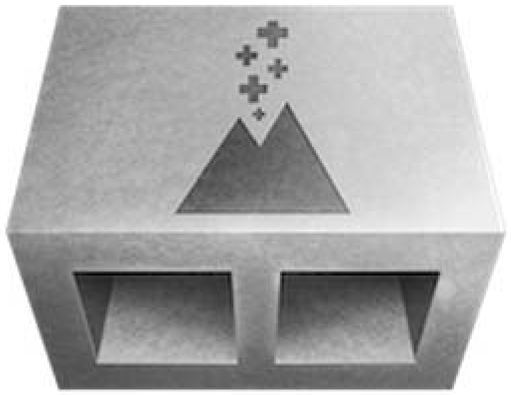
ci :: Area

cv :: Mat

cv :: Rect

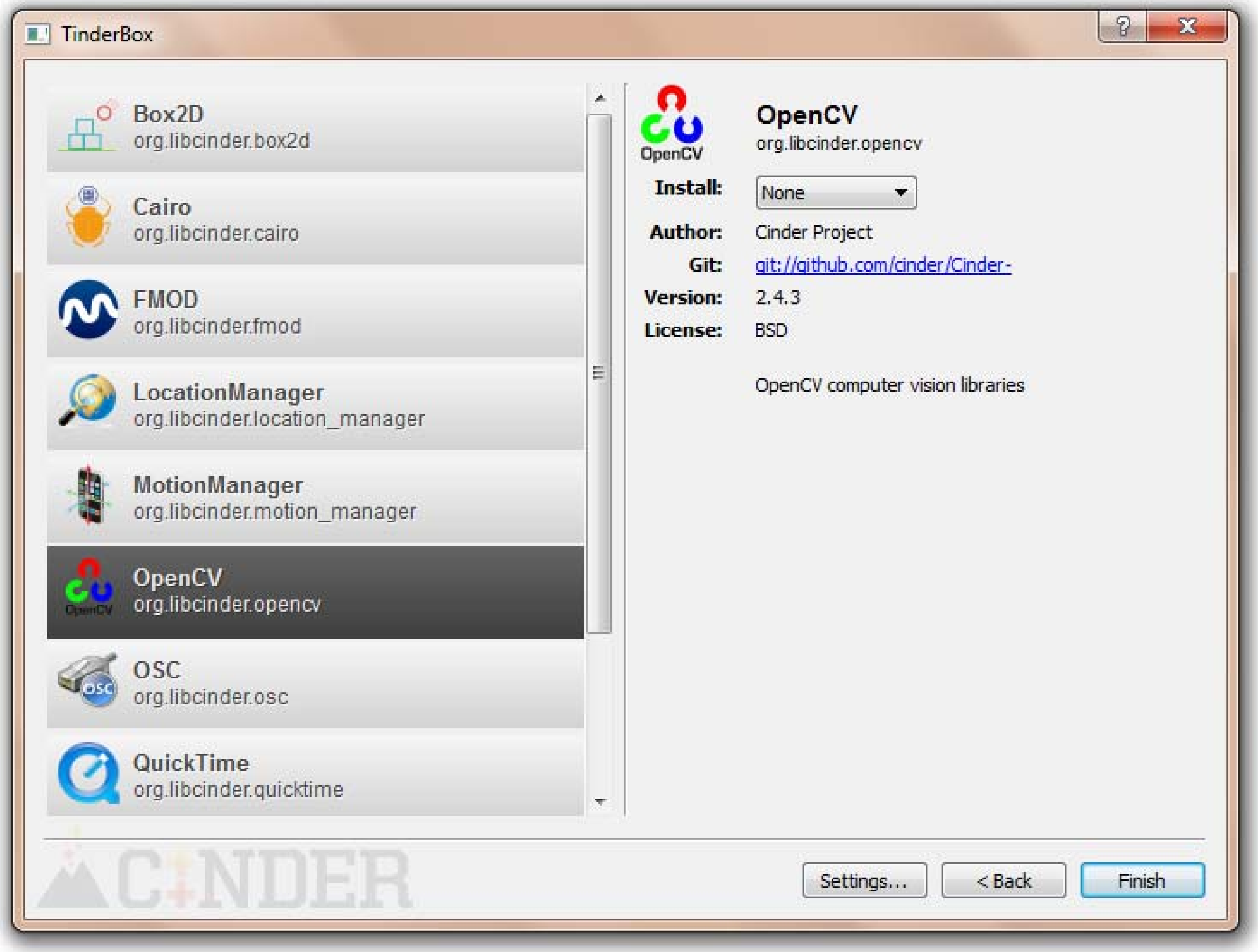
# Exchanging Data Between Cinder and OpenCV

## CinderBlocks



Prepackaged collection of code and libraries

# Selecting CinderBlocks in TinderBox



TinderBox

| | |
|---|---|
| Box2D | org.libcinder.box2d |
| Cairo | org.libcinder.cairo |
| FMOD | org.libcinder.fmod |
| LocationManager | org.libcinder.location_manager |
| MotionManager | org.libcinder.motion_manager |
| OpenCV | org.libcinder.opencv |
| OSC | org.libcinder.osc |
| QuickTime | org.libcinder.quicktime |

**OpenCV**
org.libcinder.opencv

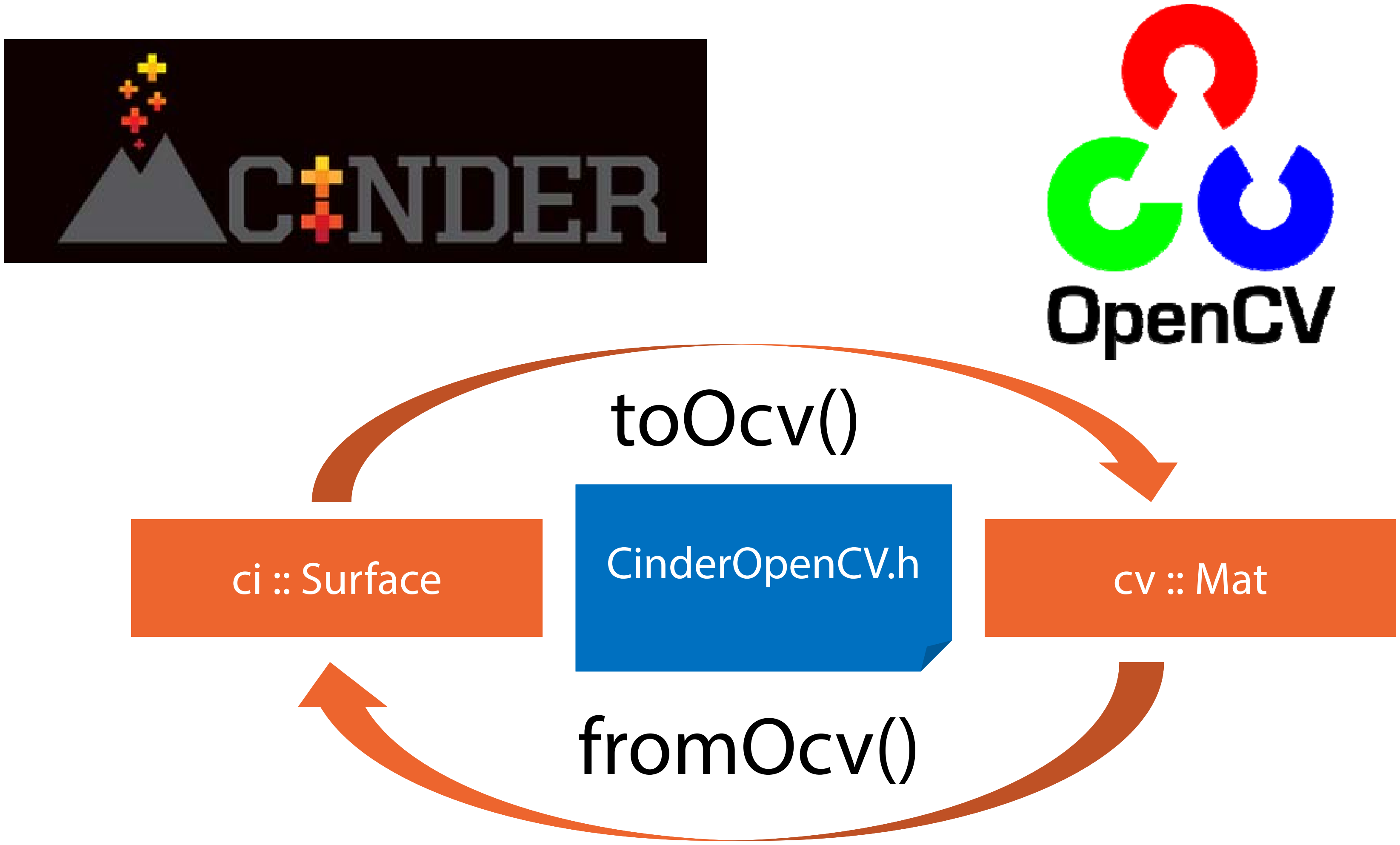
| | |
|---|---|
| **Install:** | None |
| **Author:** | Cinder Project |
| **Git:** | git://github.com/cinder/Cinder- |
| **Version:** | 2.4.3 |
| **License:** | BSD |

OpenCV computer vision libraries

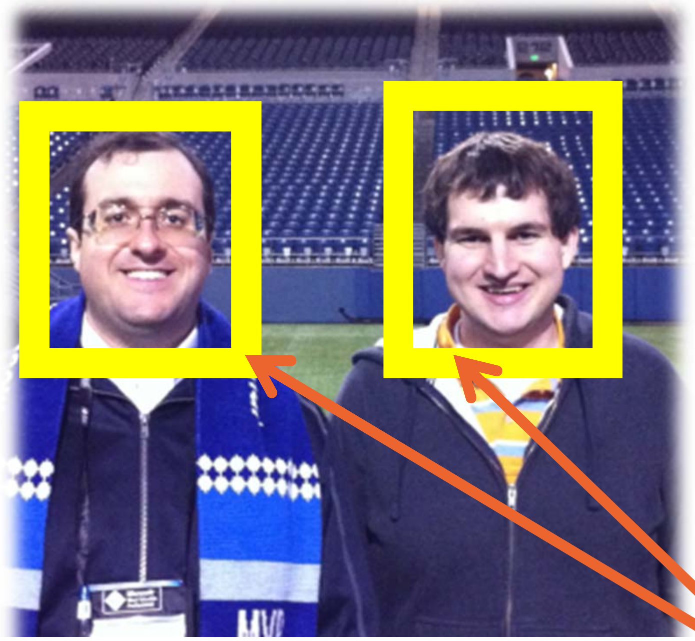
## Exchanging Data Between Cinder and OpenCV

toOcv
fromOcv

## Bridging Functions: toOcv and fromOcv



## Face Rectangles: From OpenCV to Cinder

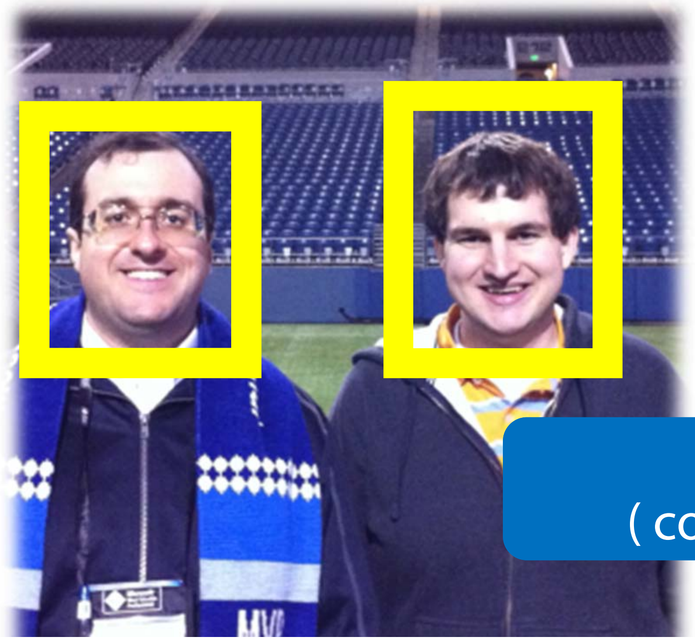


drawStrokedRect()

```
mFaceClassifier.detectMultiScale( cvImage, cvFaces );
```

std::vector< cv :: Rect >

# Face Rectangles: From OpenCV to Cinder



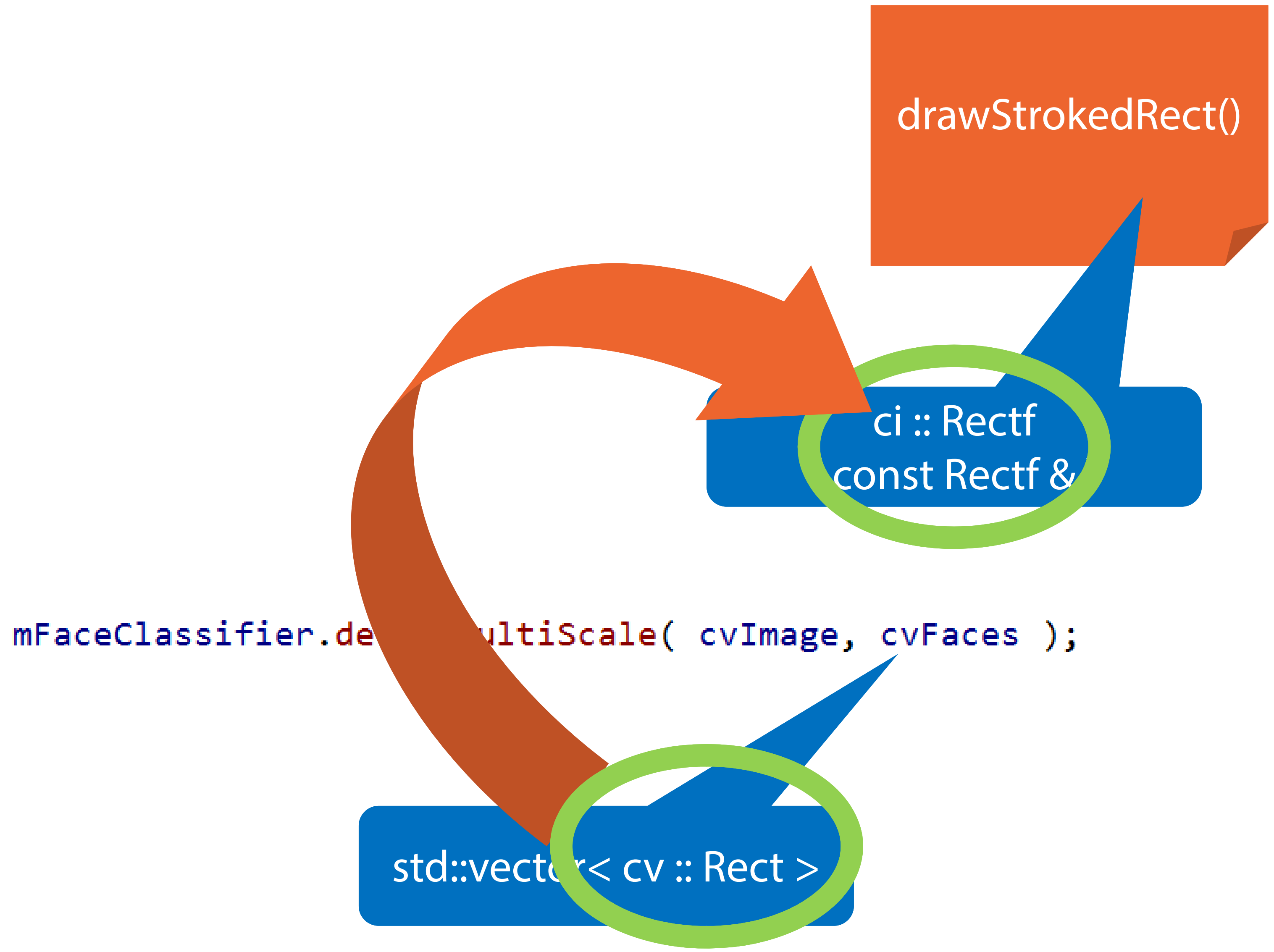
drawStrokedRect()

ci :: Rectf
( const Rectf & )

```
mFaceClassifier.detectMultiScale( cvImage, cvFaces );
```

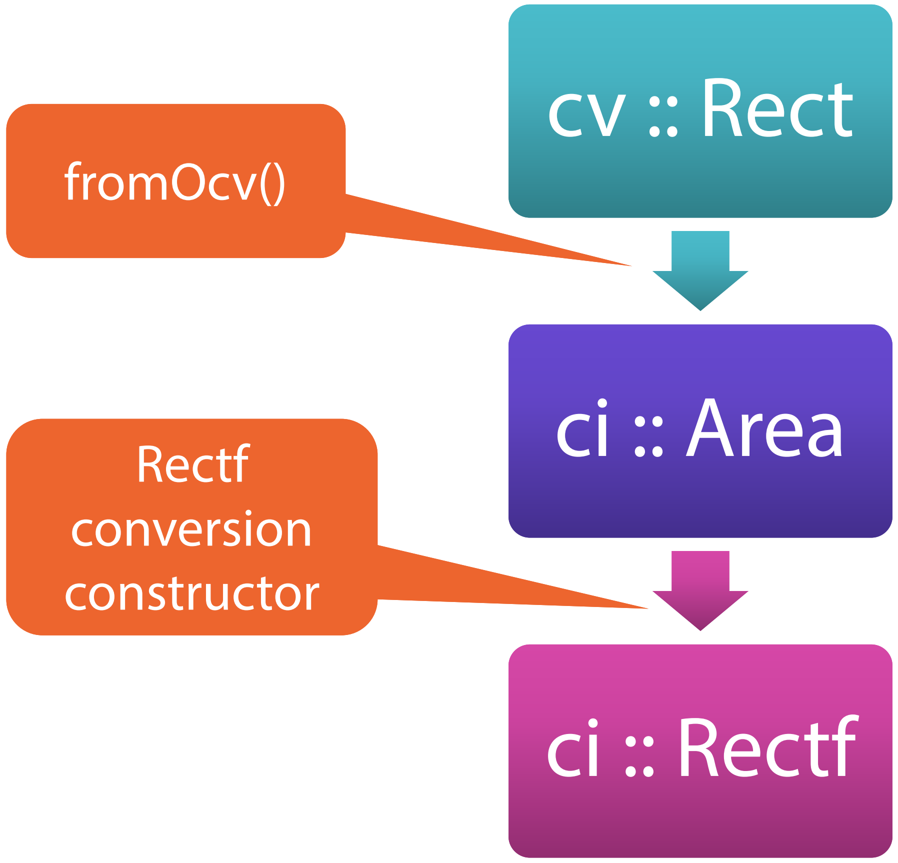
std::vector< cv :: Rect >

# Face Rectangles: From OpenCV to Cinder

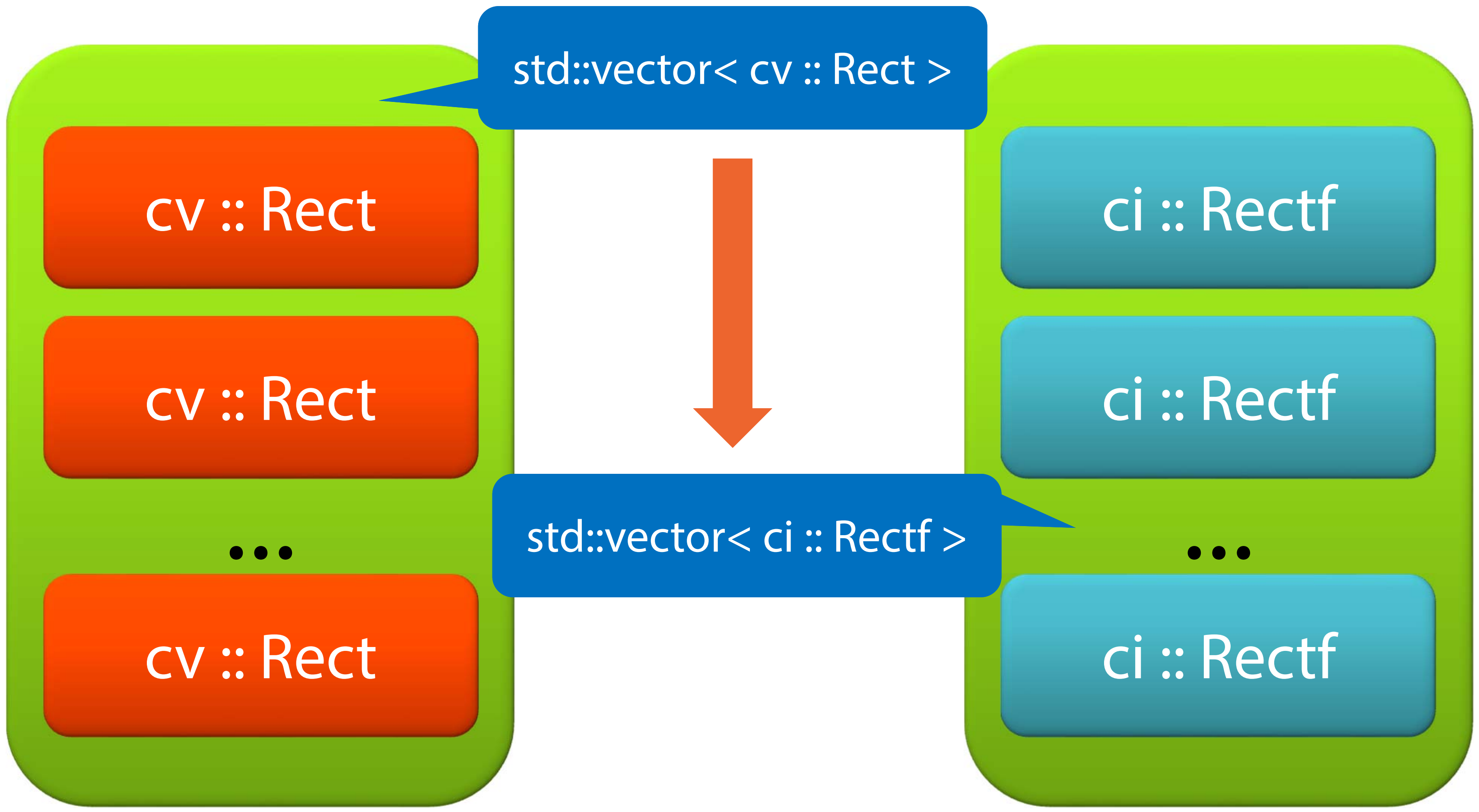


```
mFaceClassifier.de      ultiScale( cvImage, cvFaces );
```

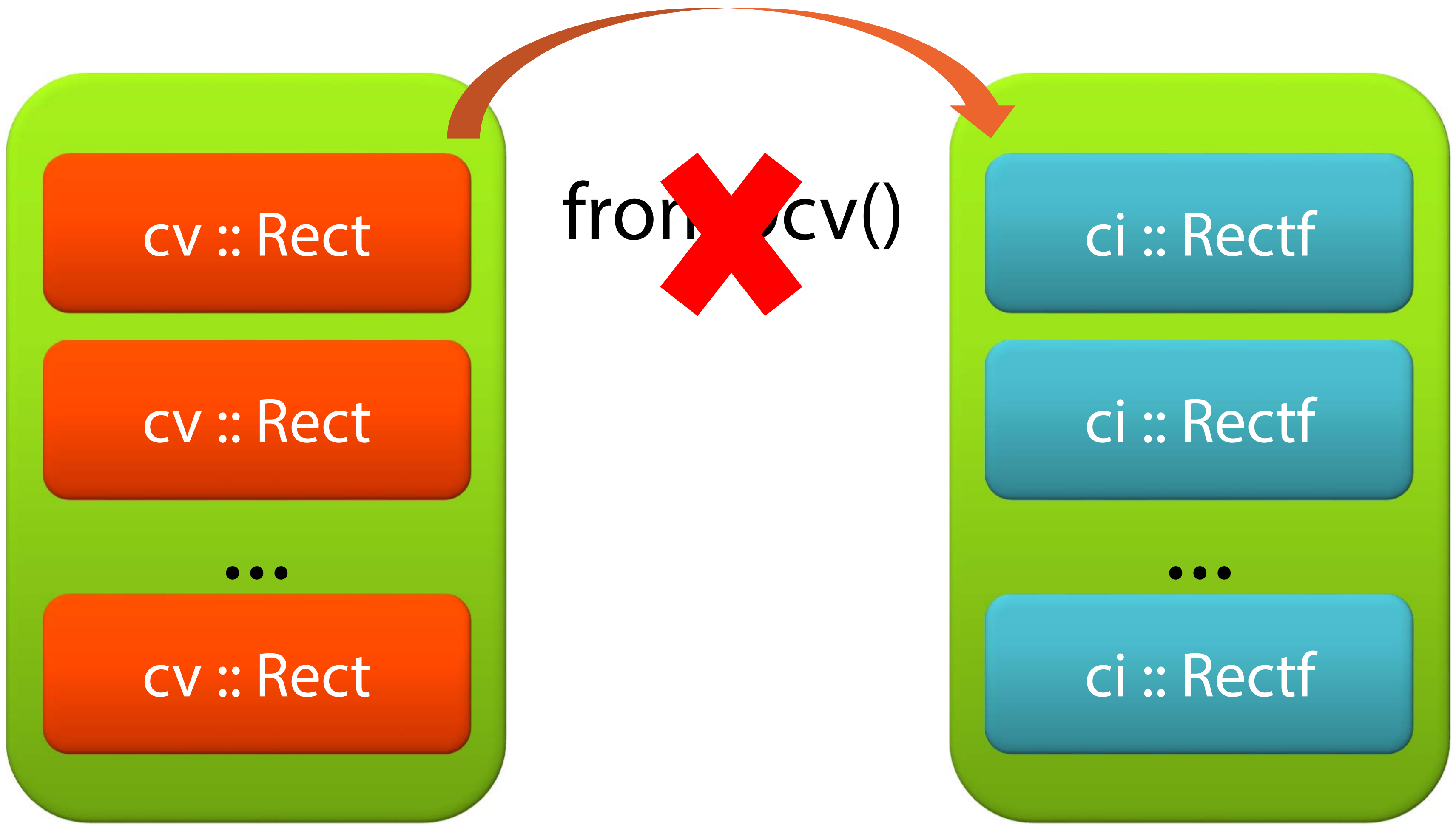
## From cv::Rect to ci::Rectf



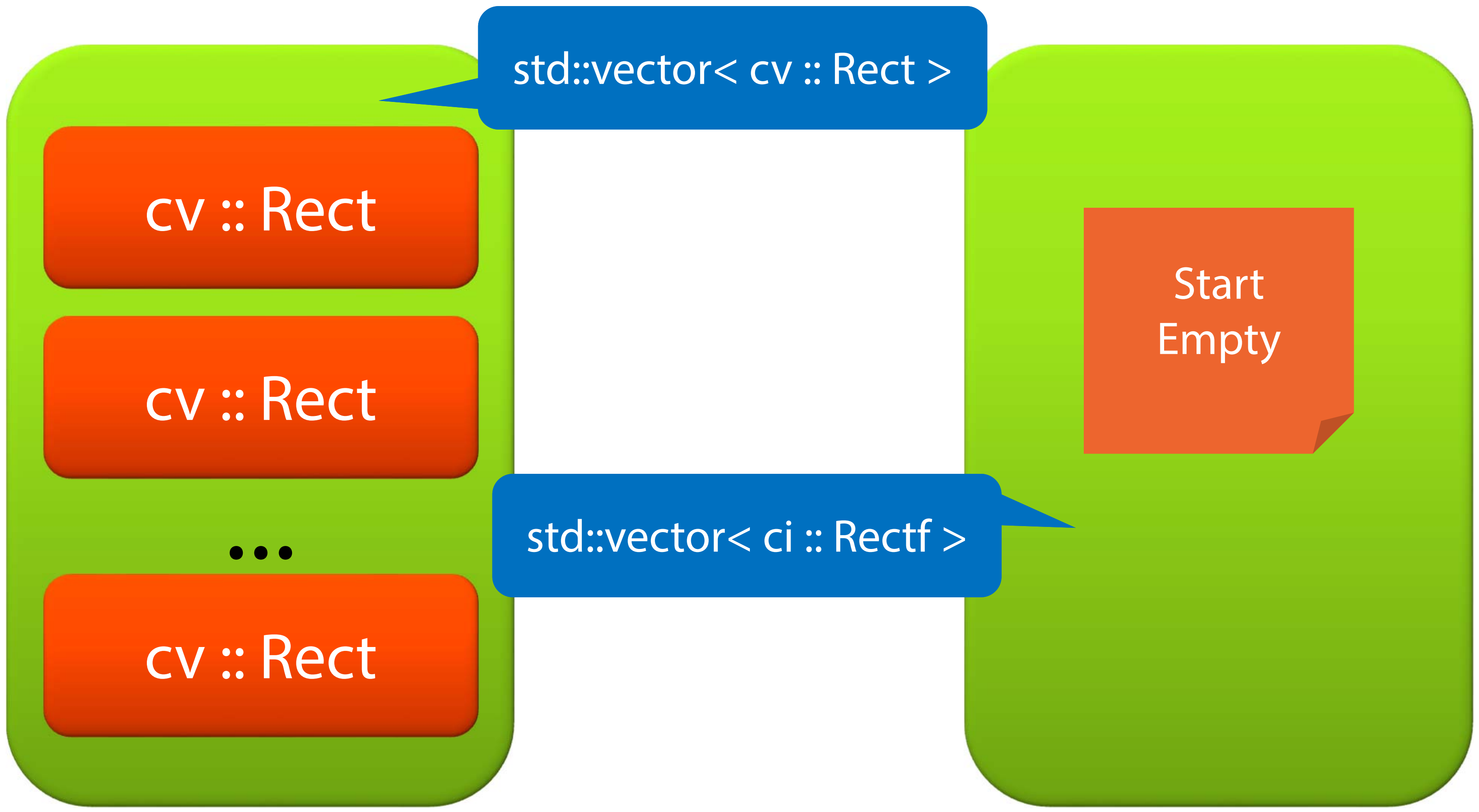
## From vector<cv::Rect> to vector<ci::Rectf>



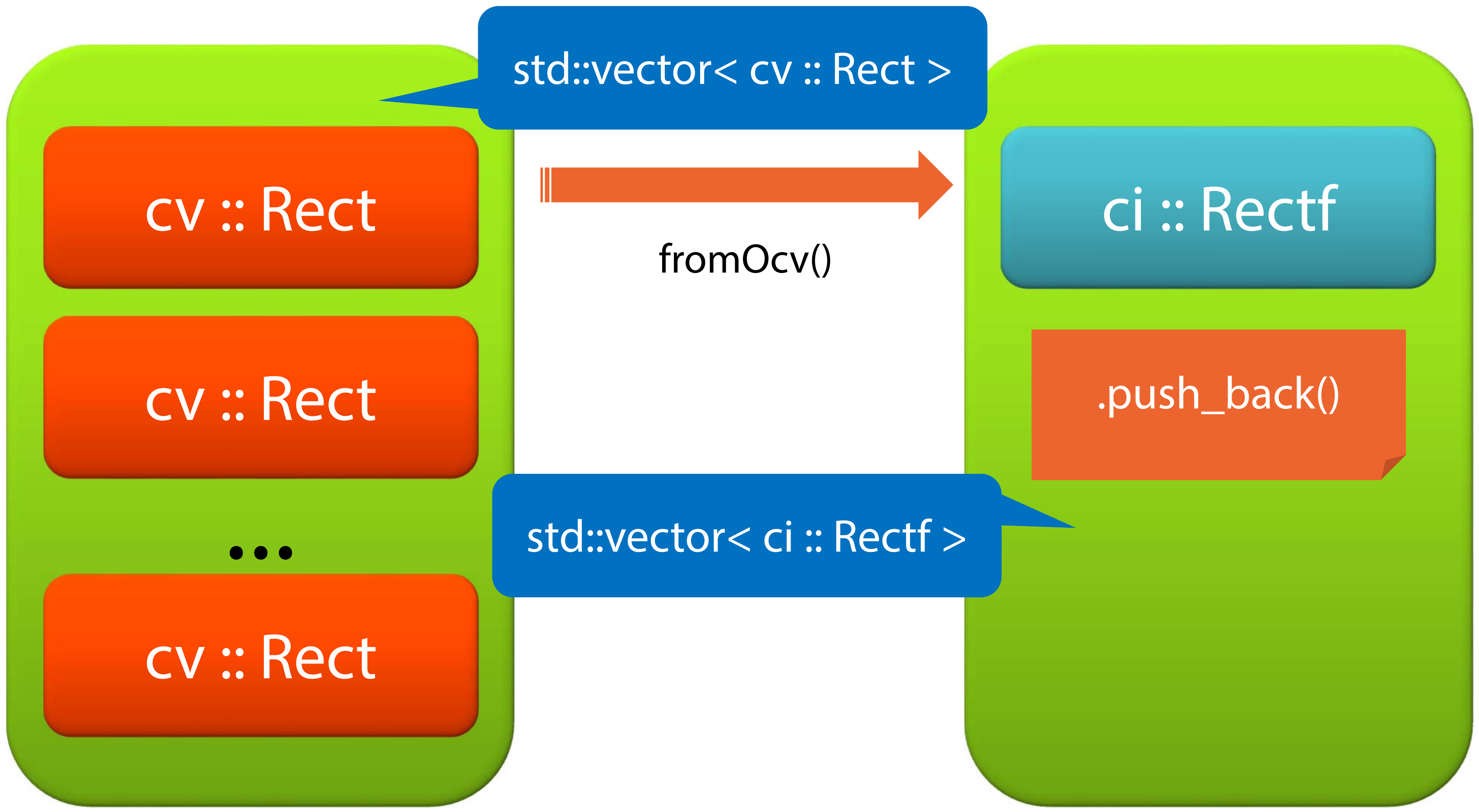
## From vector<cv::Rect> to vector<ci::Rectf>



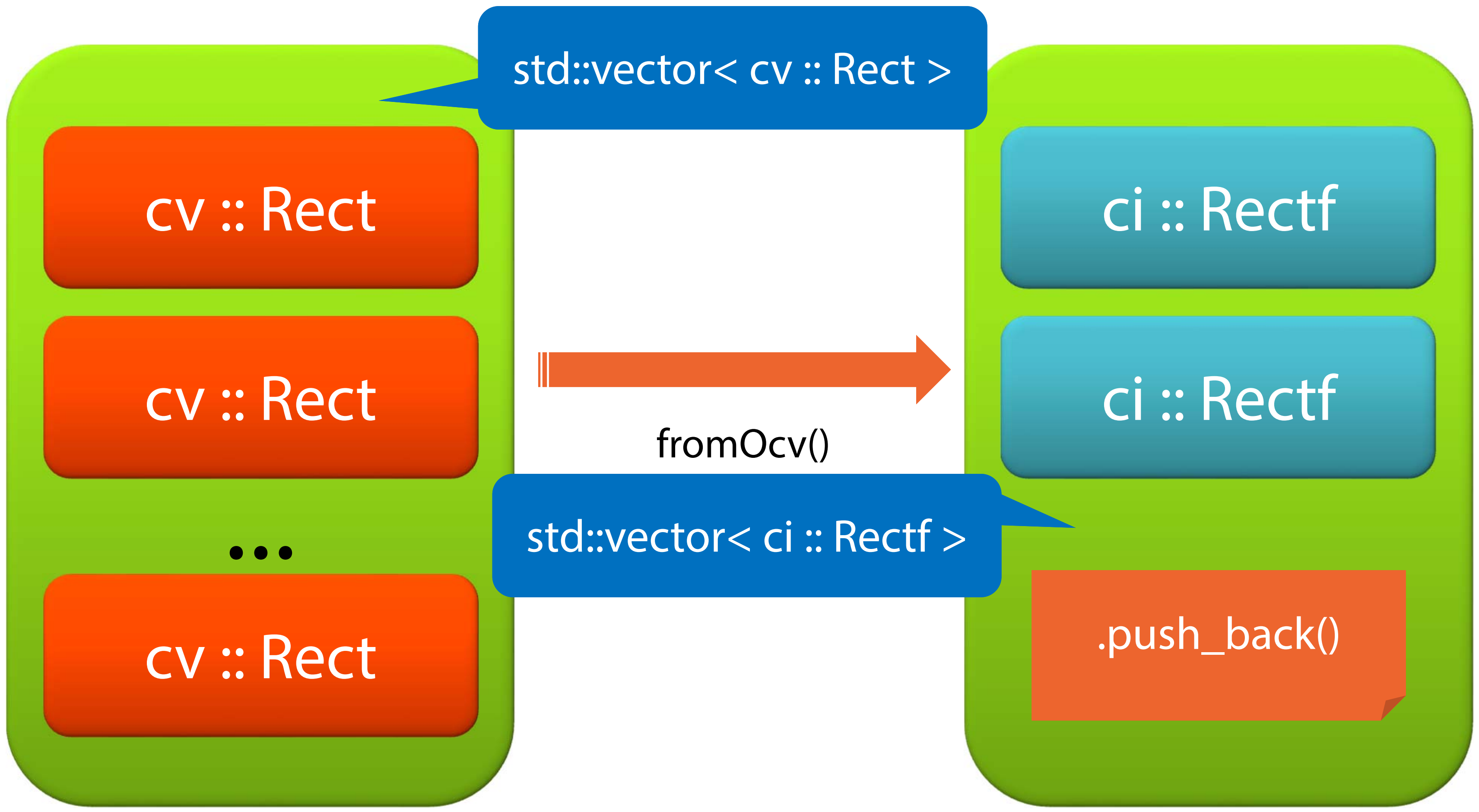
# From vector<cv::Rect> to vector<ci::Rectf>



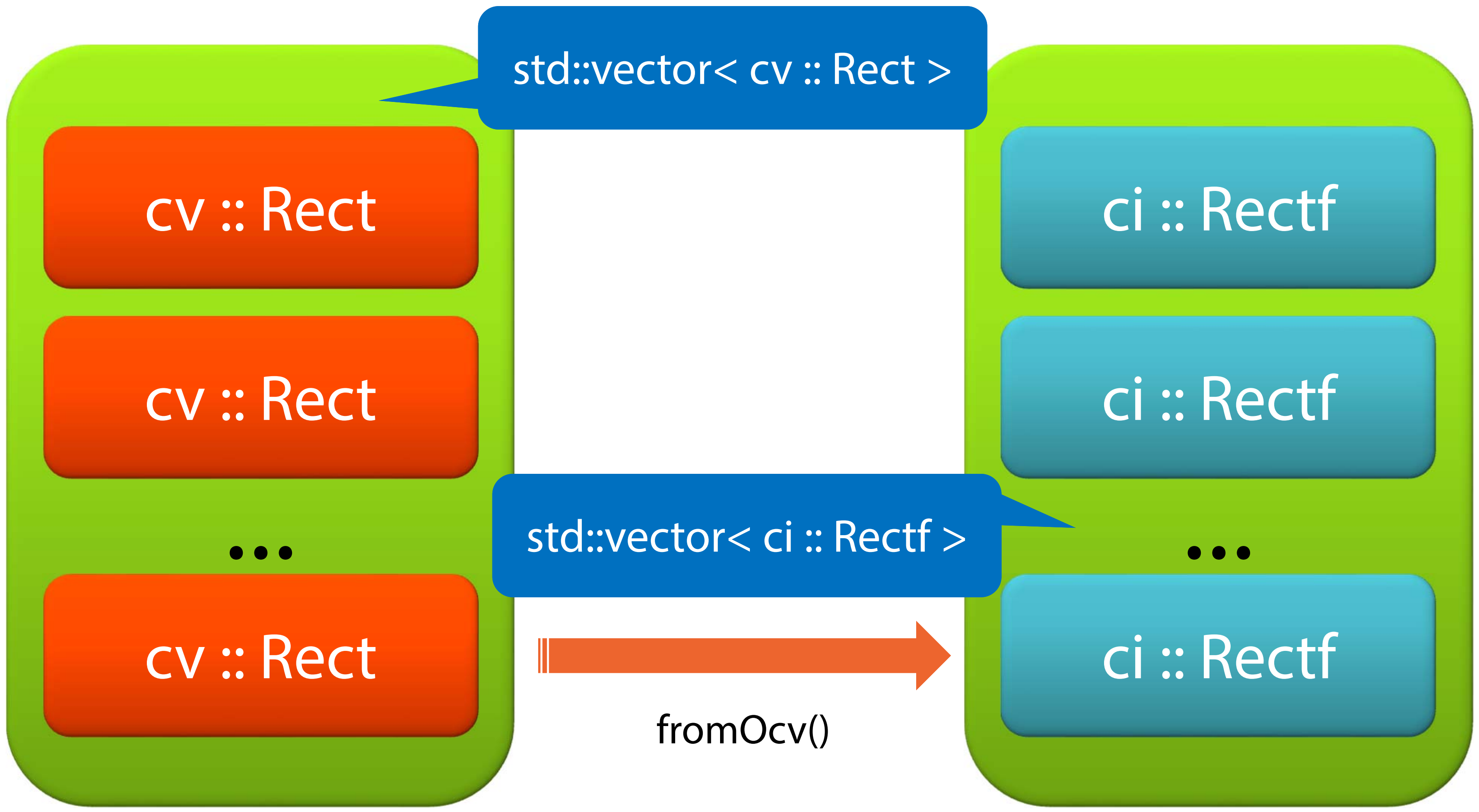
# From vector<cv::Rect> to vector<ci::Rectf>



## From vector<cv::Rect> to vector<ci::Rectf>



## From vector<cv::Rect> to vector<ci::Rectf>



## Rectangle Vector Conversion Function

```
vector<Rectf> fromOcv(const vector<cv::Rect>& cvRects) {
    vector<Rectf> rects;

    for (const auto& cvRect : cvRects) {
        Rectf rect( ci::fromOcv(cvRect) );
        rects.push_back(rect);
    }

    return rects;
}
```

## Demo: Face Detection

## Summary

- **Face detection with OpenCV**
  - CascadeClassifier
  - load()
  - detectMultiScale()
- **OpenCV/Cinder integration**
  - toOcv()
  - fromOcv()