ORACLE

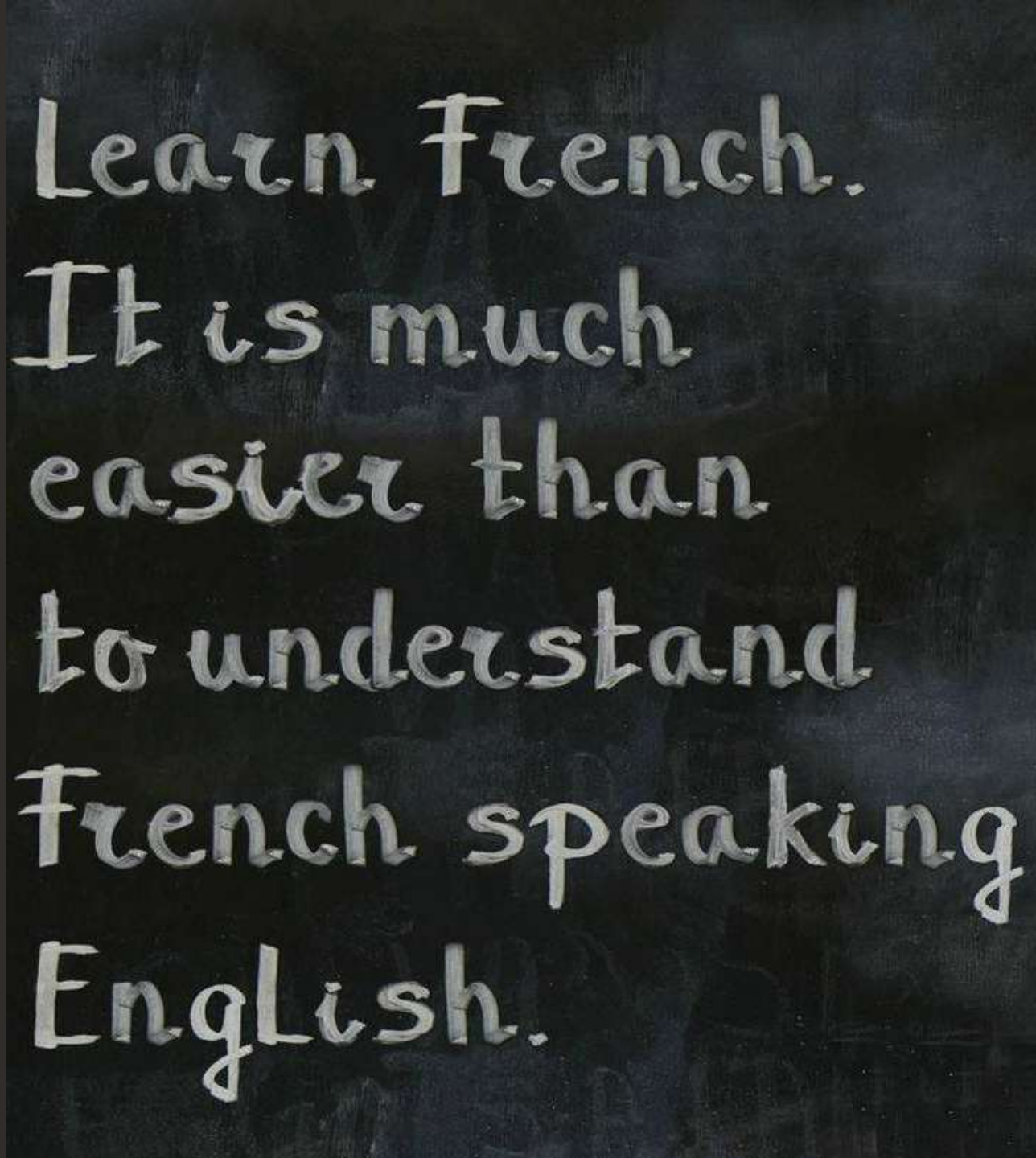
Rock, Paper, Scissors, Computer Vision

David Delabassée
@delabassee
DevRel
Java Platform Group - Oracle

ConFoo.CA

February 2020

David Delabassée
@delabassee
Java Platform Group



Safe harbor statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

Discover Computer Vision, with OpenCV, in Java.

Computer Vision?

Computer vision is an interdisciplinary scientific field that deals with how computers can be made to gain high-level understanding from digital images or videos.
From the perspective of engineering, it seeks to automate tasks that the human visual system can do.

https://en.wikipedia.org/wiki/Computer_vision

CV - Applications

- Information reading
  - OCR
  - Barcode
  - QR code
  - License plate, ...
- Inspection & verification
- Assisting humans
  - Tasks
  - Interactions, …

- Security
- Robotic
- Navigation
- Medical Imaging
- Augmented Reality
- Photography
- Video Games

...

OpenCV

- Provides a common infrastructure for CV applications
  - Inc. real-time capabilities
  - Inc. image manipulation capabilities
- Started as an internal Intel Research project
  - Advance CPU-intensive applications
  - Open sourced in 2002 (BSD)
- C++ code base with bindings
  - Python, Java, Android, JS, Swift & MATLAB
- Multiple platforms
  - Windows, Linux, macOS, Android, iOS & browsers

https://opencv.org

OpenCV - Features

+2500 optimized algorithms

- Faces detection & recognition
- Contours detections, objects identifications
- Human actions classifications
- Movements, moving objects tracking
- Images stitching
- Similar images finding
- Eye movements following
- …
- Image manipulations

OpenCV - Main Modules

• core	Core functionality
• imgproc	Image Processing
• imgcodecs	Image file reading & writing
• videoio	Video I/O
• highgui	High-level GUI
• video	Video Analysis
• calib3d	Camera Calibration & 3D Reconstruction
• features2d	2D Features Framework

OpenCV - Main Modules (cont.)

- objdetect — Object Detection
- dnn — Deep Neural Network module
- ml — Machine Learning
- flann — Clustering & Search in Multi-Dimensional Spaces
- photo — Computational Photography
- stitching — Images stitching
- gapi — Graph API

OpenCV - Extra Modules

- aruco — ArUco Marker Detection
- bioinspired — Biologically inspired vision models and derivated tools
- cnn_3dobj — 3D object recognition and pose estimation API
- cudacodec — Video Encoding/Decoding
- face — Face Analysis
- fuzzy — Image processing based on fuzzy mathematics
- hdf — Hierarchical Data Format I/O routines
- optflow — Optical Flow Algorithms
- quality — Image Quality Analysis API
- stereo — Stereo Correspondance Algorithms
- ... — +50 in total!

OpenCV

Faces Detection

Features Detection

In a Nutshell…

1. Pre-Processing

2. Processing

```
{
    "type": "tree",
    "aprxH": "16",
    "coord", "…",
    …
}
```

3. Post-processing

Pre-Processing

- Acquisition phase
  - Load an image from disk
  - Extract a single frame from a stream
  - …
- Prepare the image
  - Remove useless information (e.g. remove background)
  - Remove redundant information (e.g. lower resolution)
  - Remove noise
  - Adjust colour spaces
  - … make it easier to process!

Processing

- What to identify? Aka the **target**
- How to **identify** and uniquely **characterize** the **target**?

- Leverage specific **algorithm**(s)
- Or BYOA
  - Find useful **cues**
  - Turn cues into **evidences** to uniquely **identify** a **target**(s)

1. Cue(s) → **2. Evidence(s)** → **3. Target(s)**

Rock, paper, scissors

Rock, Paper, Scissors

- Input — Picture of a hand
- Output — Label corresponding to 1 gesture, out of 3
- Environment — Controlled
  - Lighting
  - Background
  - Pose
  - Skin
  - No jewelry
  - No polydactyly

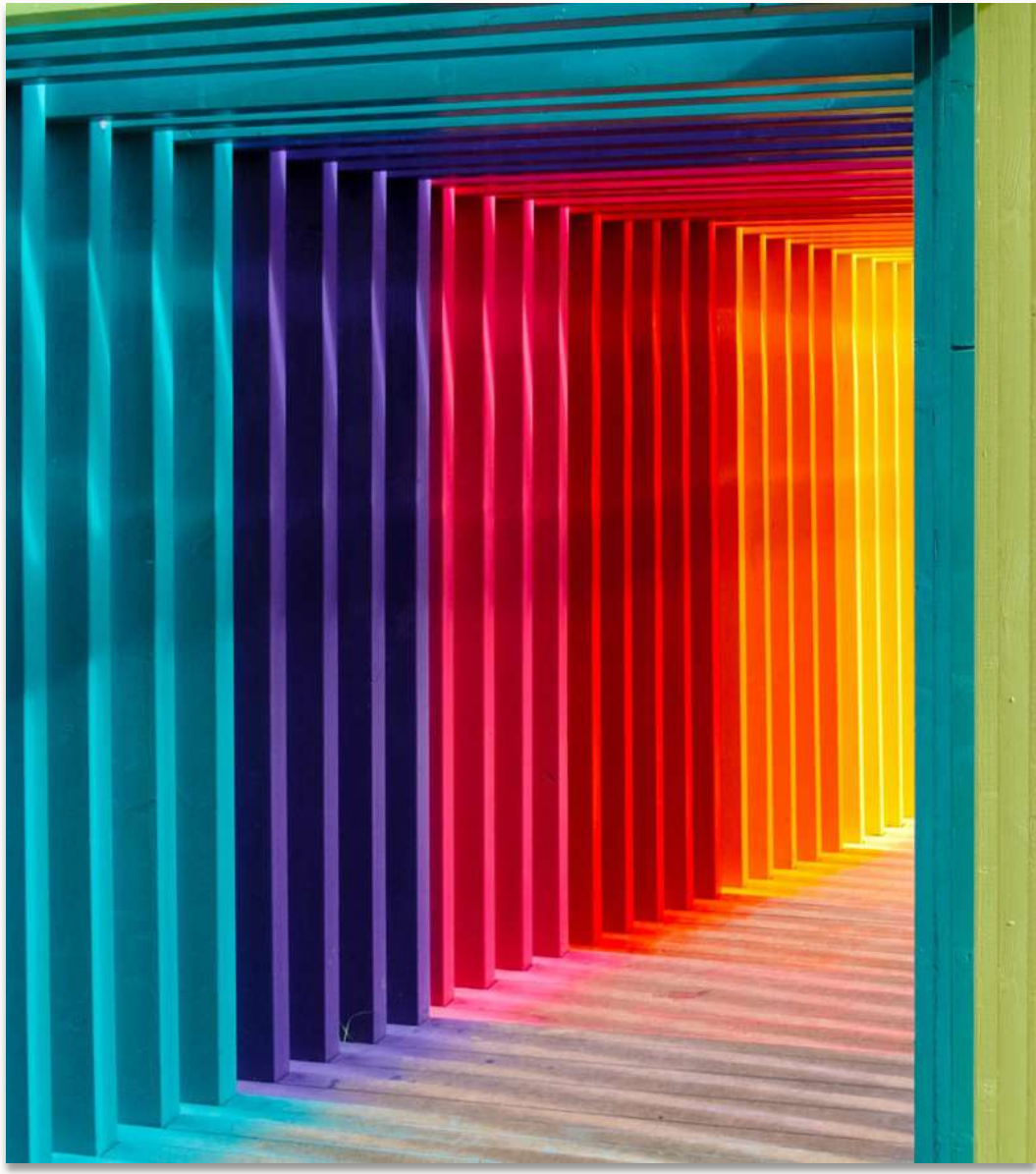
https://en.wikipedia.org/wiki/Rock_paper_scissors

OpenCV Features

- Color Spaces
- Blurring
- Thresholding
- Contour Finding
- Convex Hull
- Edge Detection
- GrabCut
- …

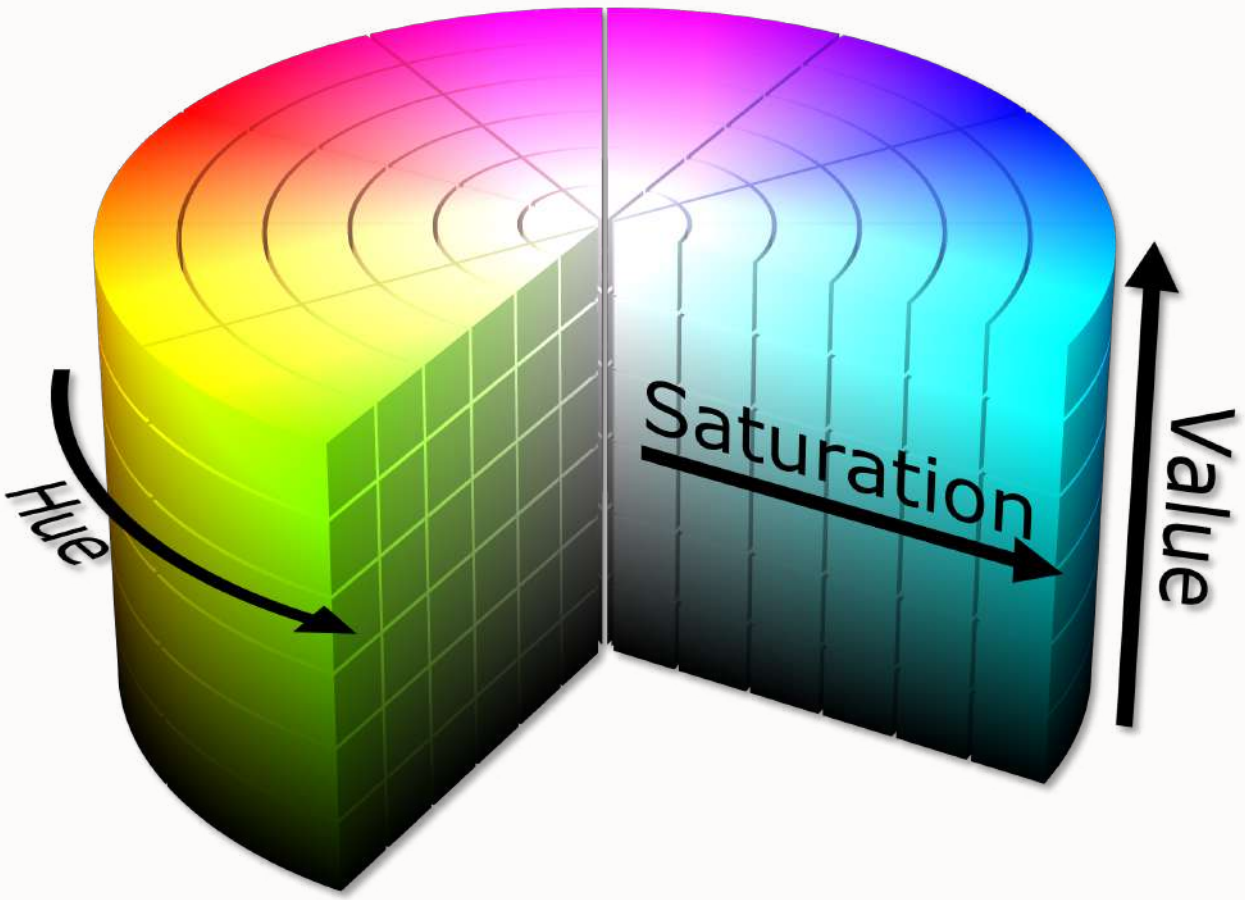
Color Spaces

- Way of organizing colors
- Absolute or generic
- RGB, CMYK additive, Pantone, NCS, HSL, etc.
- OpenCV uses BGR by default



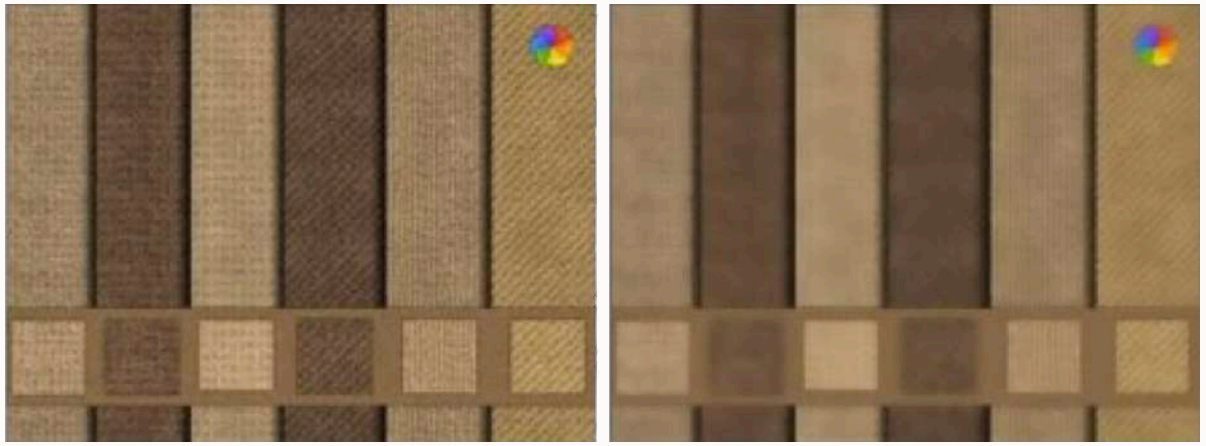
HSV

- Hue
  - 0° - 360° : Red
  - 60° : Yellow
  - 120° : Green
  - 180° : Cyan
  - 240° : Blue
  - 300° Magenta
- Saturation
  - 0 - 100%
- Value
  - 0 - 100%



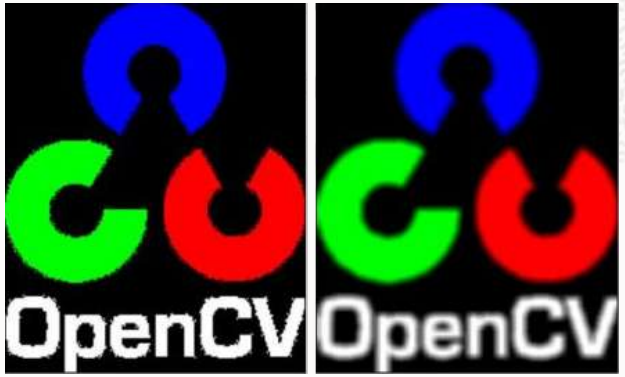
Blurring

- Removes high frequency content
  - Ex. Noise, edge, etc.
- Averaging (1)
- Gaussian (2) & median blurring (3)
- Bilateral filtering (4)

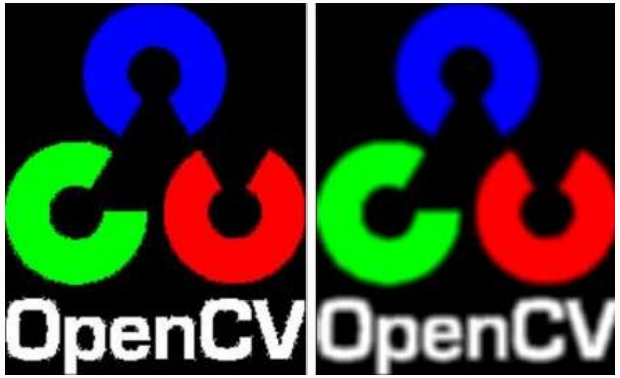


(4)

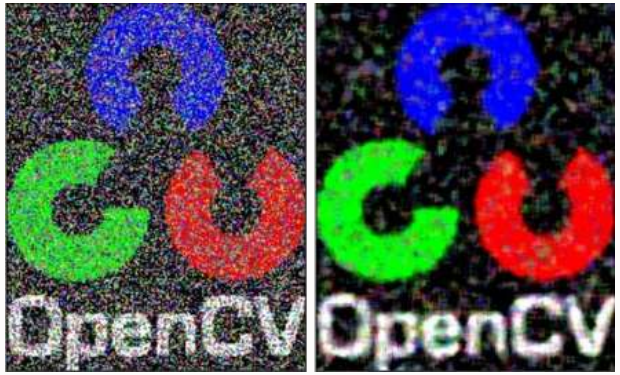
(1)



(2)

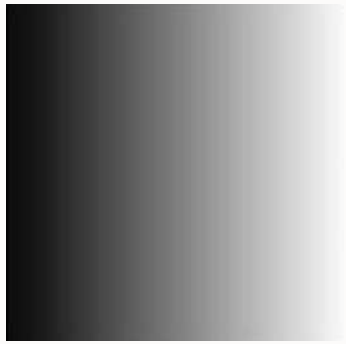


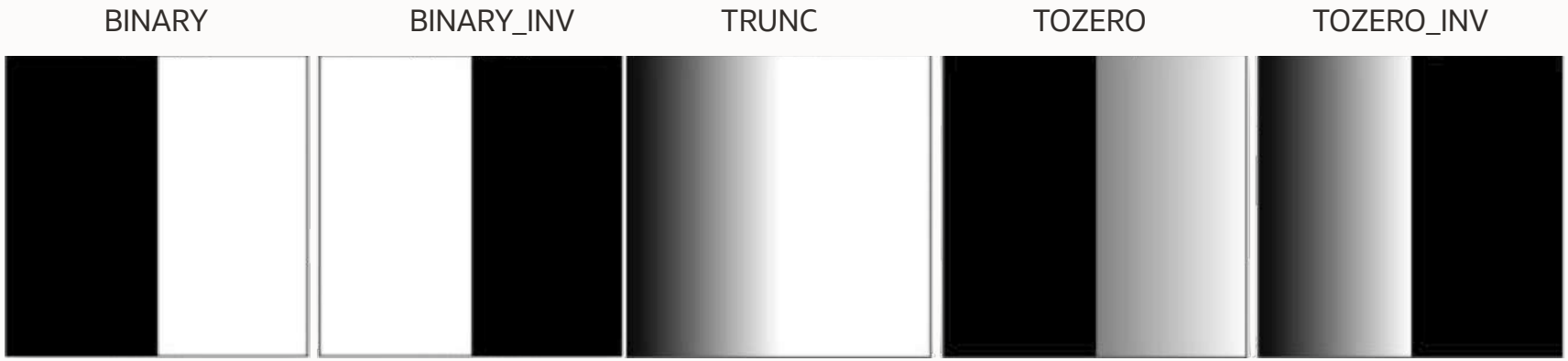
(3)



Thresholding

- If the pixel value is smaller than the threshold, it is set to 0, otherwise it is set to a maximum value



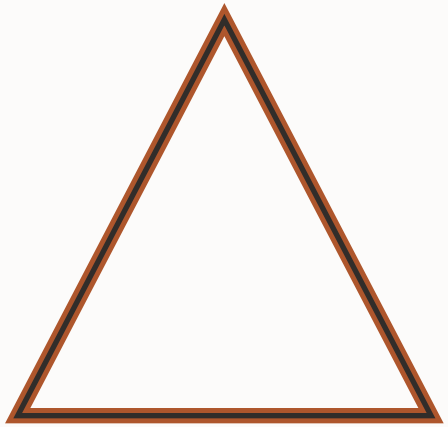


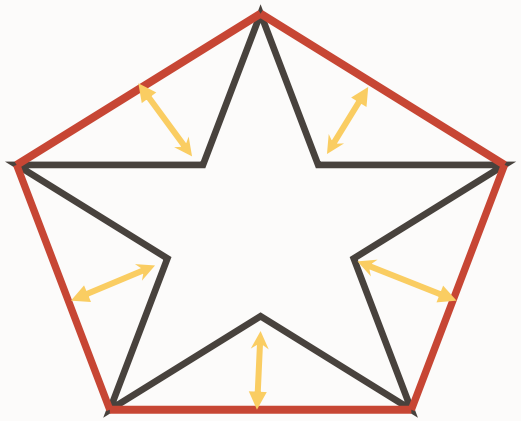
Contours

- Curve joining all the continuous points along some boundary
- Easier on “simple” images
  - E.g. binary images
  - Threshold, canny edge detection, etc.
- A contour has a size

Convex Hull

- Minimal convex set wrapping a set of points
- One or more convexity defect(s) (opt.)
- Algorithms
  - Jarvis Gift Wrapping
  - Kirkpatrick's Prune and Search
  - Chan
  - …





OpenCV & Java

- Build your own OpenCV distribution
  - Platform dependent build with required modules
  - JNI wrapper
- JavaCV
  - https://github.com/bytedeco/javacv
- Java types
  - `org.opencv.core.Mat`

Hello World

Rock, paper, scissors?

Locate the hand

Human Skin Detection Using RGB, HSV and
Models

S. Kolkur[1], D. Kalbande[2], P. Shim

[1] Department of Computer Engineering, Th

Fast and E

Department of Electrical and

Abstract— In this paper, an efficie
proposed. The algorithm is based
pre-processing step utilizing the concept
conversion in order to identify candidate
subsequently, a novel local two-stage diffu

A New and Improved Skin Detection Method Using

rch Laboratory.

†

Hu
way, NJ, 08902, USA

Skin detection is the process of finding skin-colored pixels and regions in an image or a video. This process is typically used as a preprocessing step to find regions that potentially have human faces and limbs in images. Several computer vision approaches have been developed for skin

Springer Link

Published: 17 November 2018

A survey on skin detection in colored images

Sinan Naji, Hamid A. Jalab & Sameem A. Kareem

Artificial Intelligence Review **52**, 1041–1087(2019) | Cite this article

580 Accesses | **3** Citations

Abstract

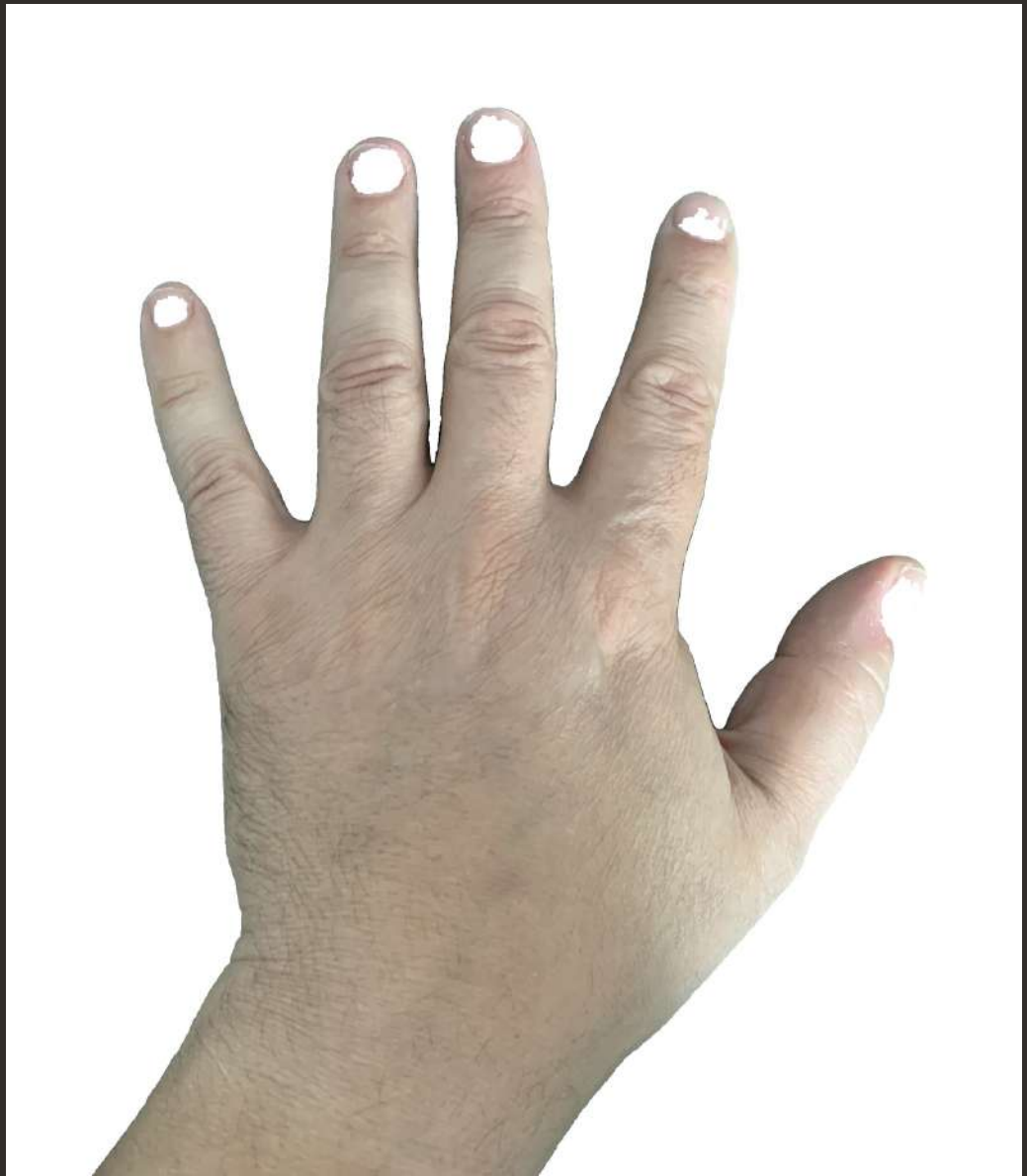
Color is an efficient feature for object detection as it has the advantage of being invariant to changes in scaling, rotation, and partial occlusion. Skin color detection is an essential

Locate the hand

Locate the hand

- Skin Detection
- GrabCut
- …

```
Imgproc.calcHist(hue…
average(hue…
Imgproc.threshold(huePlane, averageHue…
```



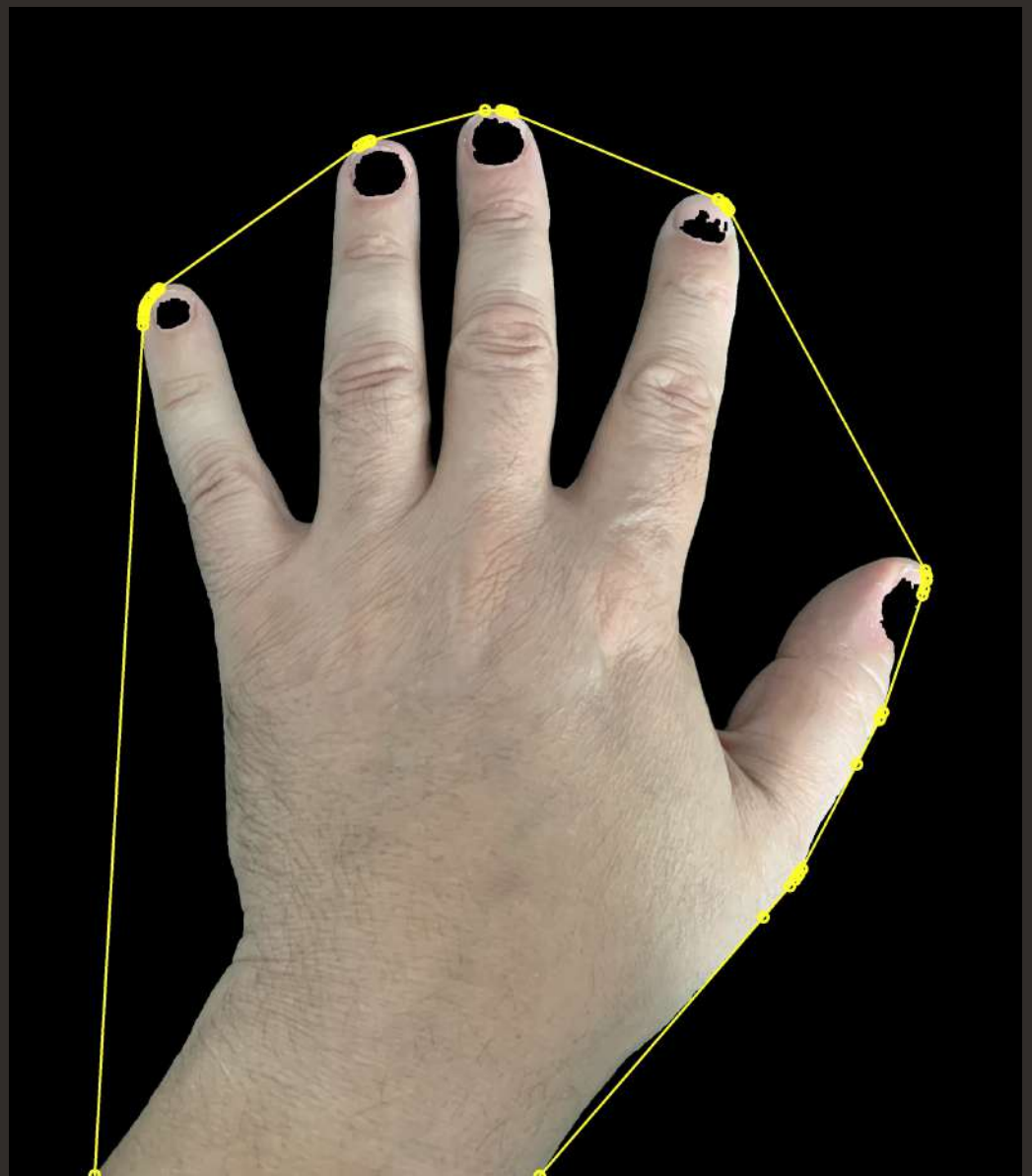
Locate the hand

Contours

```
Imgproc.findContours(…)
```

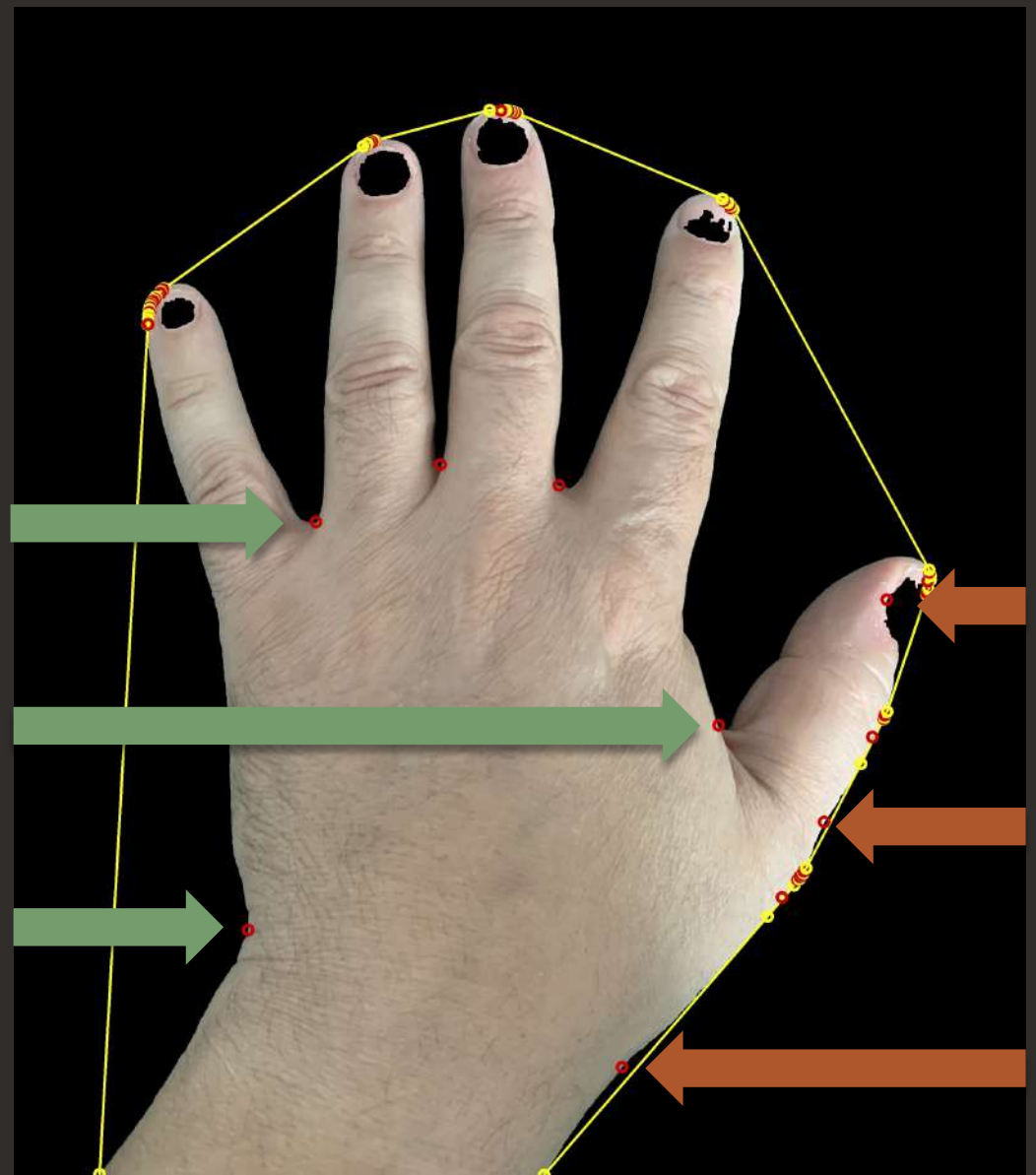
Convex Hull

```
Imgproc.convexHull(lgstContour, …)
```



Convex Hull

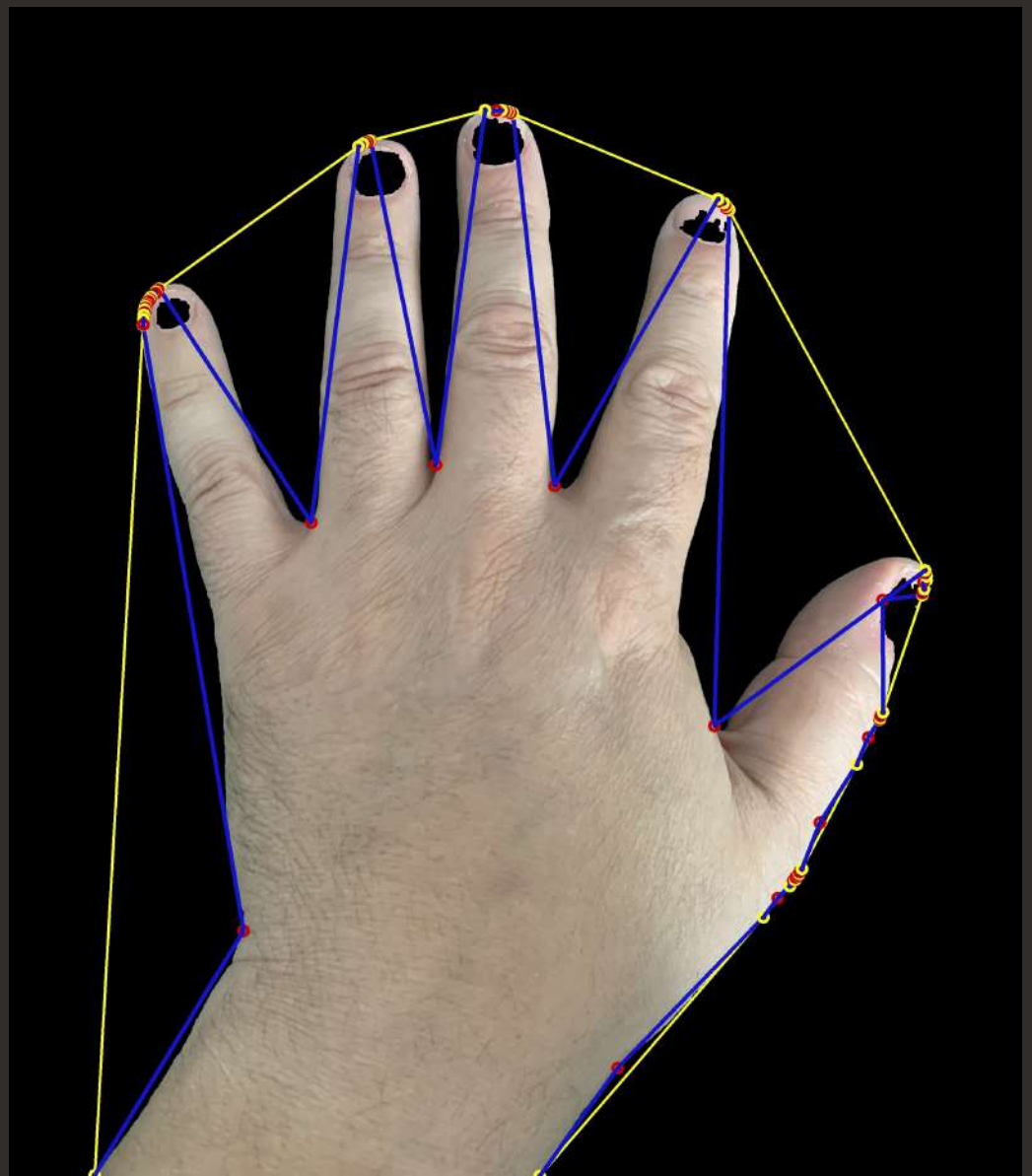
```
Imgproc.convexHull(lgstContour, …)
```



Convex Hull Defects

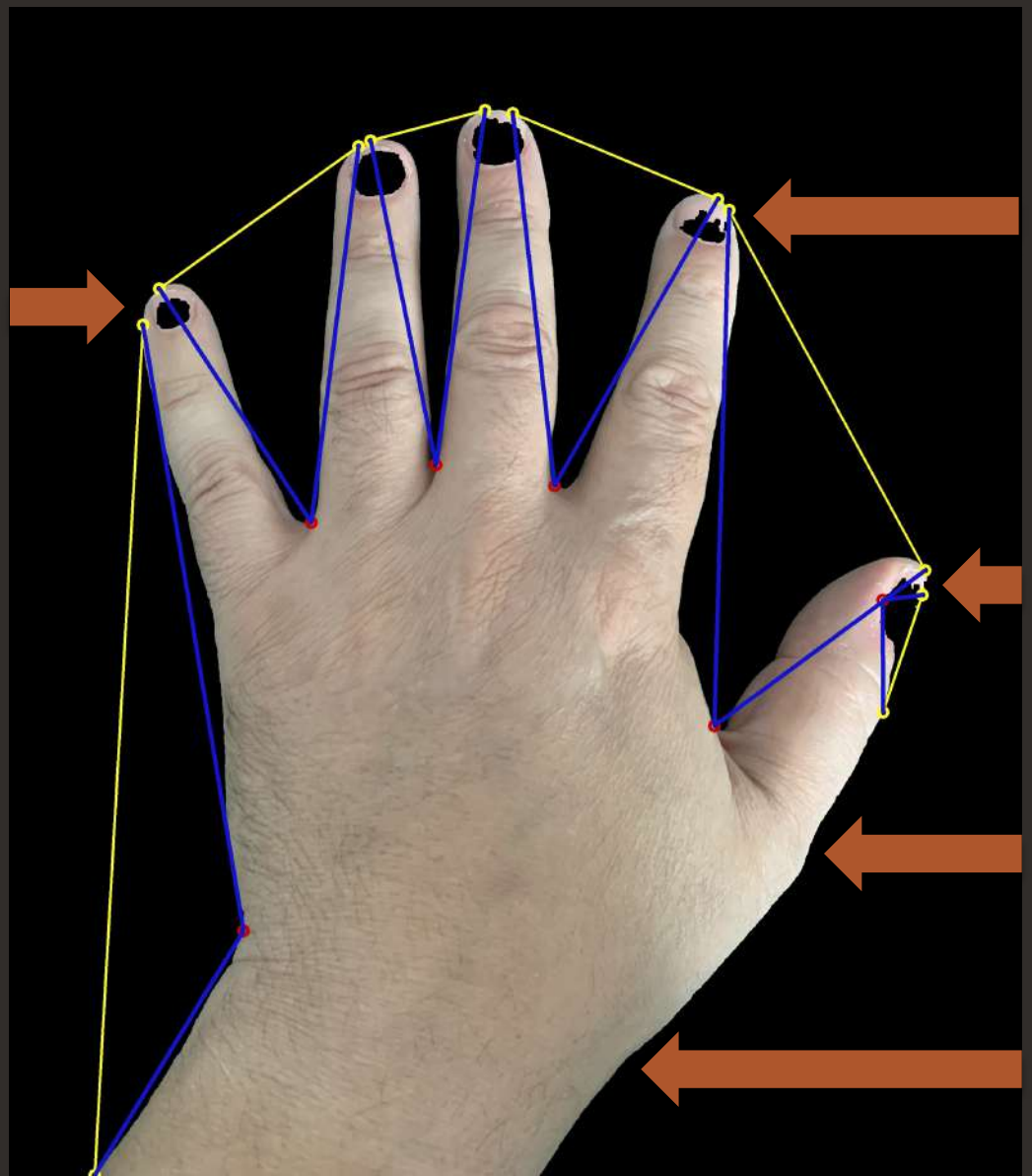
```
Imgproc.convexityDefects(lgstContour,
hull, …)
```

- StartPoint
- EndPoint
- CenterPoint



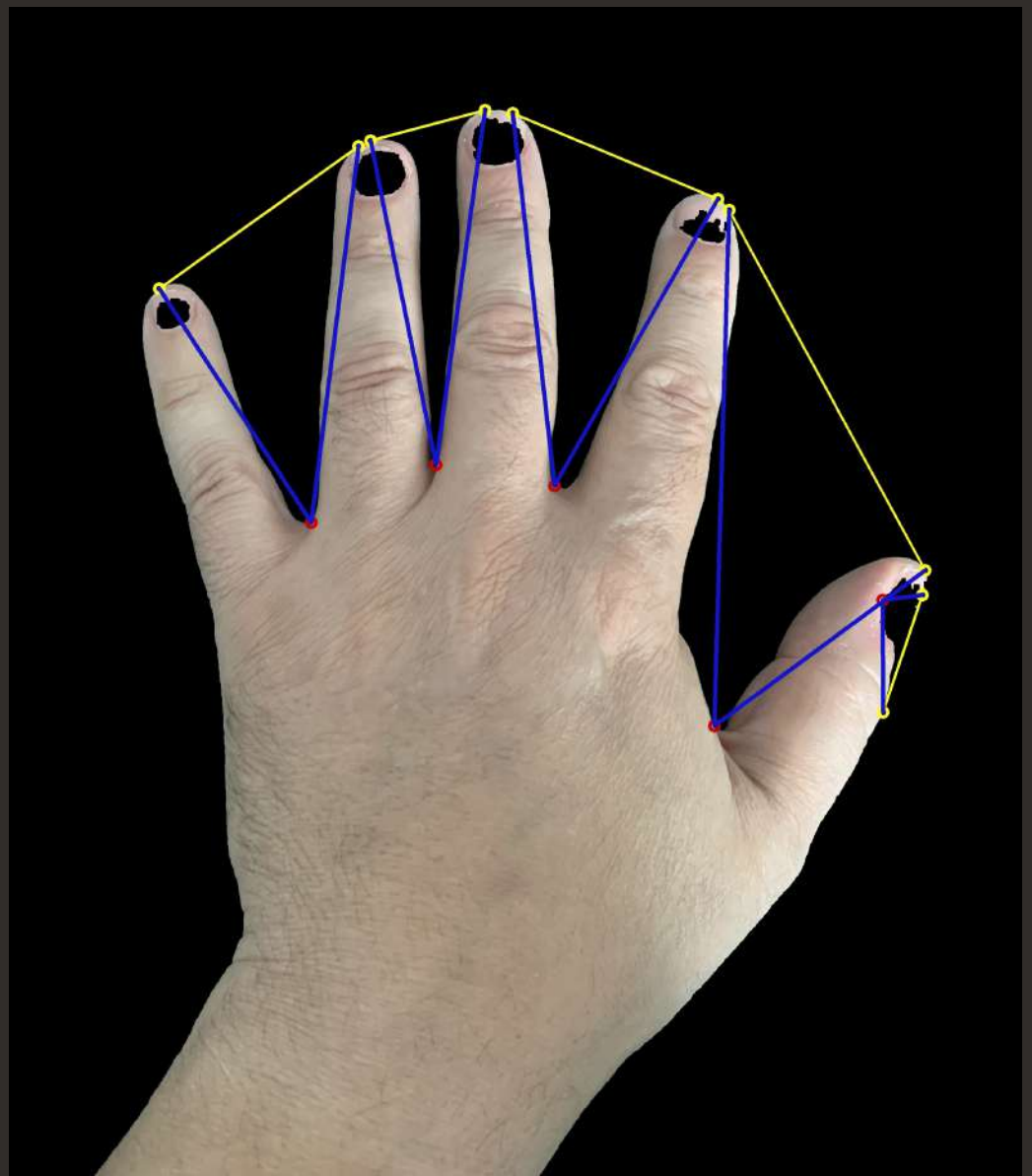
Convex Hull Defects

```
convexDefects
    .stream()
    .filter(cvx -> cvx.getDepth > 10000)
      .forEach(cvx -> {...});
```



Convex Hull Defects

```
convexDefects
    .stream()
    .filter(cvx -> cvx.getDepth > 10000)
      .forEach(cvx -> {...});
```

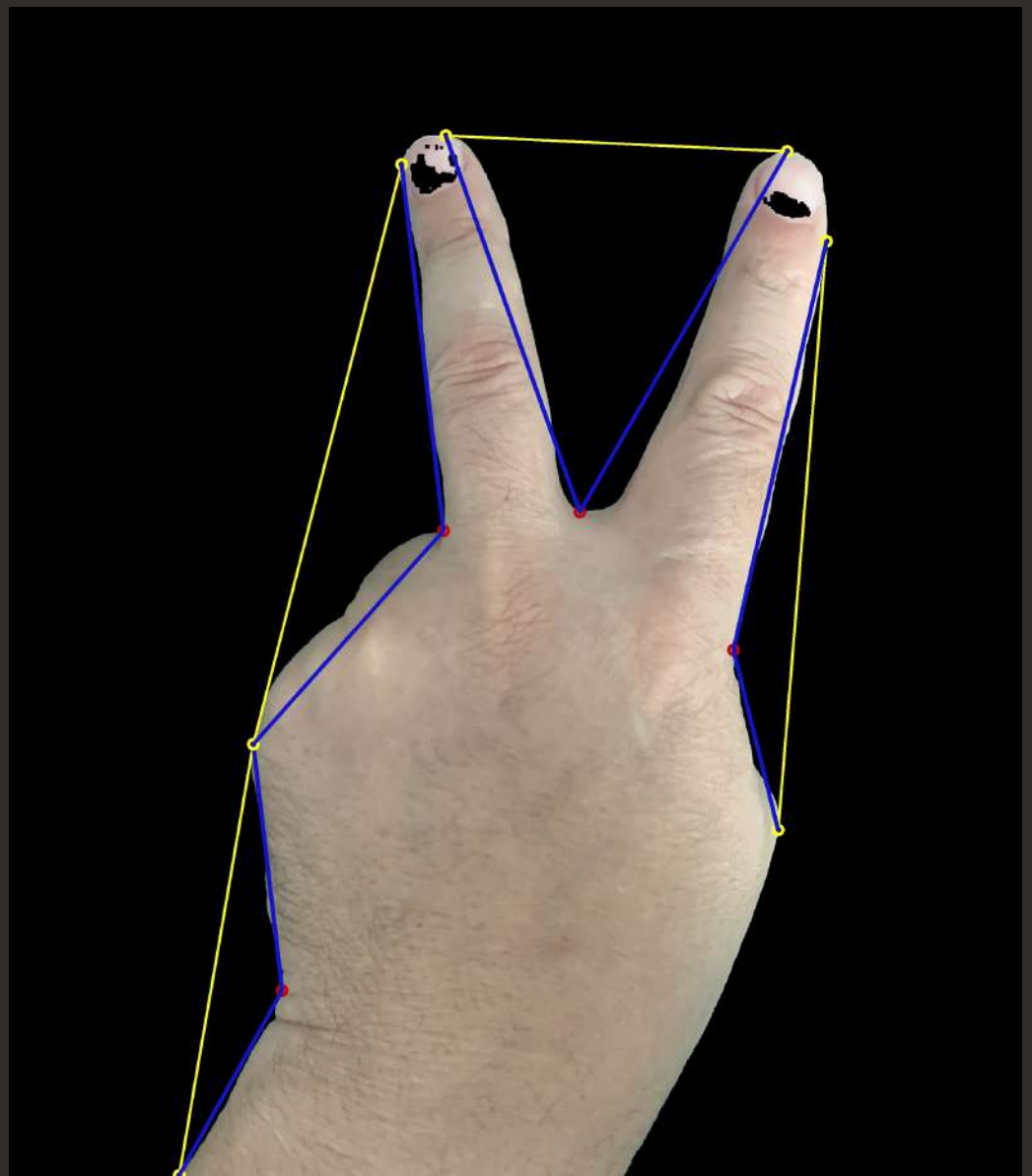


Rock, paper, scissors?

```
convexDefects
  .stream()
  .filter(cvx -> cvx.getDepth() > 10000)
  .filter(cvx -> cvx.getAngle() < 110)
  .forEach(cvx -> {...});
```

⇨ paper!

Rock or scissors?



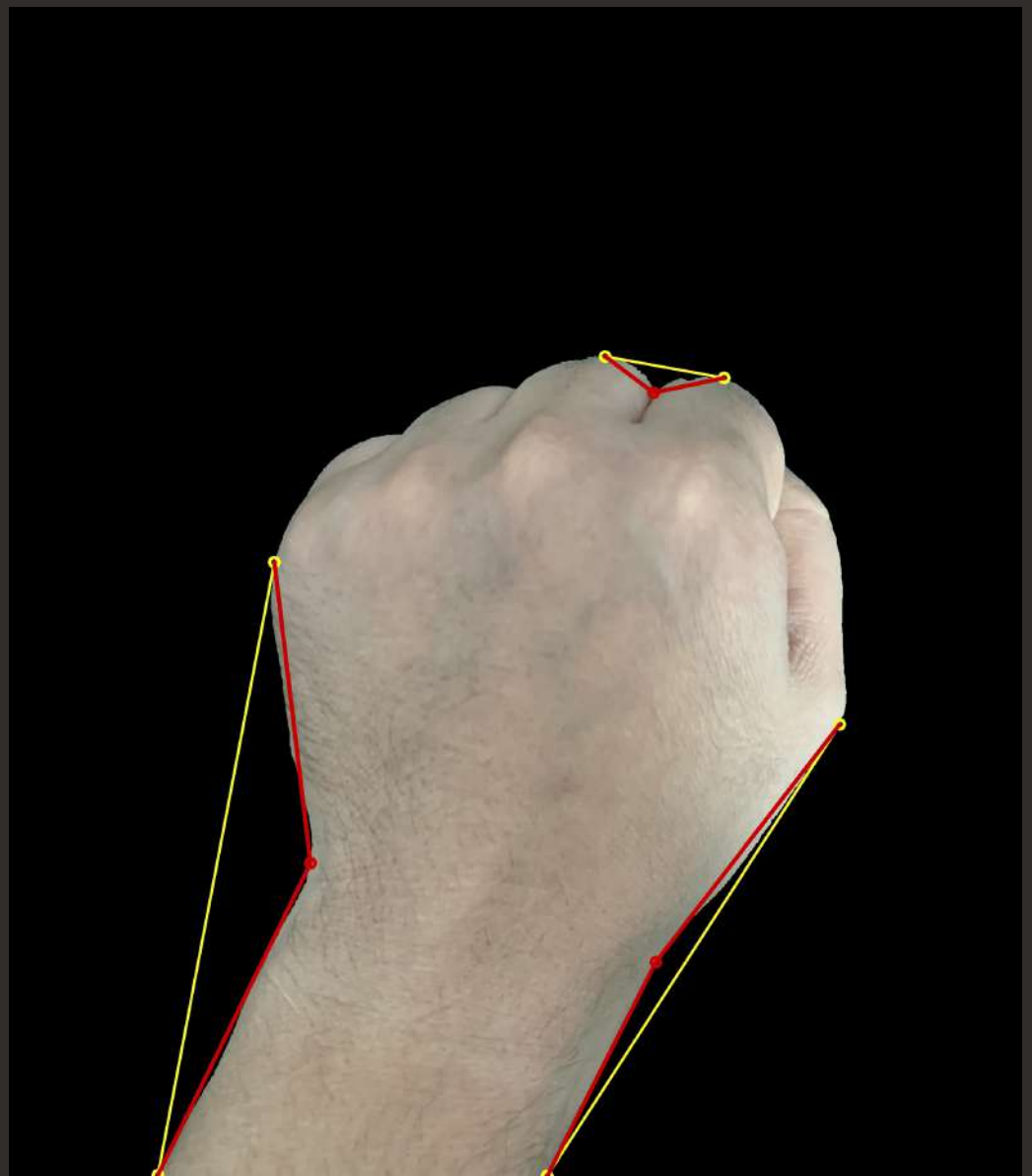
Rock or scissors?

```
convexDefects
  .stream()
  .filter(cvx -> cvx.getDepth() > 10000)
  .forEach(cvx -> {...});
```

⇨ scissors!

Rock?

```
convexDefects
   .stream()
   .filter(cvx -> cvx.getDepth() > 10000)
   .forEach(cvx -> {...});
```



Rock?

```
convexDefects
  .stream()
  .filter(cvx -> cvx.getDepth() > 10000)
  .filter(cvx -> cvx.getAngle() < 110)
  .forEach(cvx -> {...});
```

⇨ rock!

Rock, paper, scissors

Computer Vision, with OpenCV, in Java.

OpenCV

- Rich capabilities, multi-platform & open-source
- Rich Ecosystem
  - GPU, OpenCL, CUDA
  - Intel IntelliSense
  - ROS, OpenCV for Unity, OpenCV plugin for Unreal Engine…
  - OpenPose, JavaCV, SimpleCV, …
  - OpenCV Hardware Partnership Program
    - Advance the development of interoperable smart vision devices
  - OpenVisionCapsules
    - Open format to facilitate the creation of portable algorithm “capsules”
  - …
- The Computer Vision library!

https://opencv.org/about

OpenCV ~~rants~~ notes

- Features, features & features!
  - Algorithms, algorithms & algorithms
- Docs
  - https://docs.opencv.org/3.4/df/d0d/tutorial_find_contours.html
  - https://docs.opencv.org/4.1.1/javadoc/index.html
  - Books

OpenCV ~~rants~~ notes

- OpenCV is written in C++
  - Bring your own builds
  - Debugging on the native side? Logging?
  - Think twice before crossing the (JNI) bridge!
- Consistency

```
public void aMethod(Mat src, Mat dest, ...)
public void anotherMethod(Mat dest, Mat src, ...)
```

Rock, paper, scissors

- Custom basic algorithm : Cue(s) ⇨ Evidence(s) ⇨ Target(s)
- Controlled environment
- Constraints on
  - the environment
  - the target(s) to identify
  - the vision
  - …

Rock, paper, scissors - RFEs

- Improve algorithms, hand detection & RPS gestures
- Video Vs. single frame
- Depth camera support
- Correct hand orientation
- Extract more information, e.g. palm detection
- Add more gestures, e.g. counting
- …
- Controlled environment vs. the “Wild West”

Thank you

@delabassee

ORACLE