TECHNISCHE UNIVERSITÄT DRESDEN

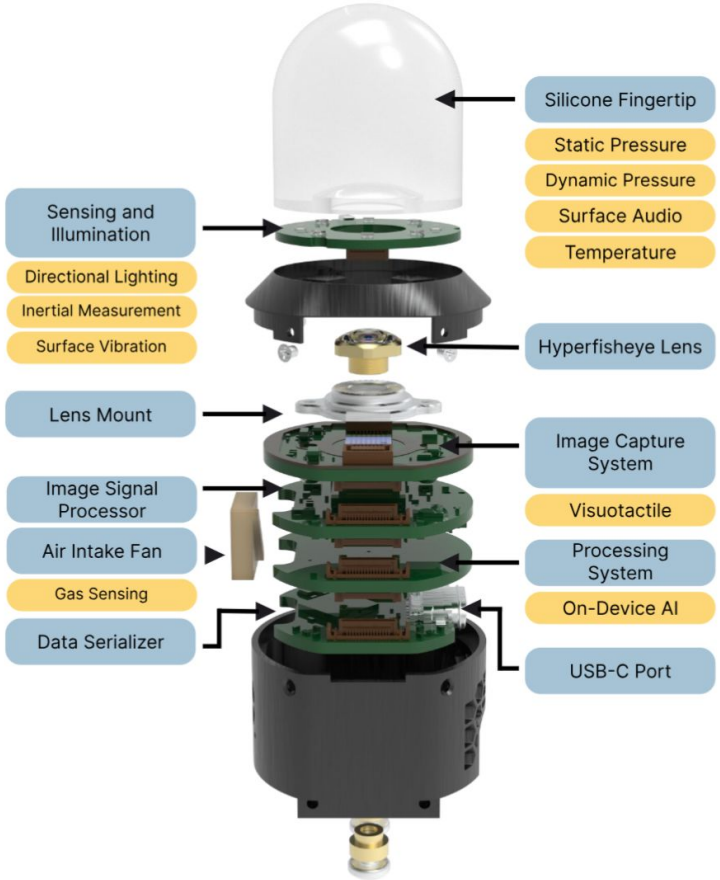
DRESDEN concept

Touch Sensing and Processing Summer School 2025

Tutorial

Tutorial // OpenTouch Interface
August 20, 2025

Touch Sensors



Digit

`get_frame()`

≠

Digit360

`cv2.VideoCapture.read()`

≠

GelSight Mini

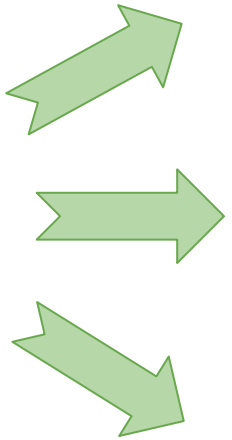
`read_frame()`

OpenTouch Interface

The interface:

```
connect()              # Connect to the sensor
disconnect()           # Disconnect from the sensor
read(modality)         # Read a given modality
set()                  # Change sensor settings
get()                  # Get sensor settings

start_recording()      # Start the data collection
stop_recording()       # Stop the data collection
```



Digit

Digit360

GelSight Mini

Pinky 2.0

Your custom sensor (?)

Option 1: People with Linux (Ubuntu)

1. Install the tools and try the sensor

(1) Install Opentouch Interface

(2) Grab a DIGIT and play around

2. Create a CNN for classification

(3) Collect datasets (*e.g., coins*)

(4) Clean datasets (*e.g., remove low quality images*)

(5) Build and train a CNN

(6) Test your model

Option 2: People with MacOS

1. Try out the sensor

(1) Use a camera to see the DIGIT's output

(2) Play around with it

2. Create a CNN for classification

(3) Download datasets from GitHub

(4) Clean datasets (*e.g., remove low quality images*)

(5) Build and train a CNN

(6) Test your model

OpenTouch Interface

1. Create a virtual Python environment

```
$ cd <a-directory-of-your-choice>
$ python -m venv venv
$ source venv/bin/activate
```

2. Install OpenTouch Interface

```
$ pip install opentouch-interface
$ opentouch-dashboard
```

Resources

Repository with Jupyter Notebook, training data and solutions

```
$ git clone git@github.com:lasr-lab/tspss-2025-tutorial.git
```

Structure of repository

```
tspss-2025-tutorial/
├── manual.pdf                    # This file
├── coin_data.zip                 # Sample datasets (2€, 1€, 50 Cent; uncleaned)
├── ubuntu/                       # For Ubuntu users
│   ├── classifier.ipynp          # <- Start here for CNN
│   ├── filter.ipynp              # <- Start here for grayscale filter
│   └── sample_solution.ipynp
└── macos/                        # For macOS users
    └── ...
```