MATLAB EXPO

May 28, 2024 | Beijing

探索光影魔术：
MATLAB数字成像与显示技术创新

Qian Zhang, MathWorks

MathWorks

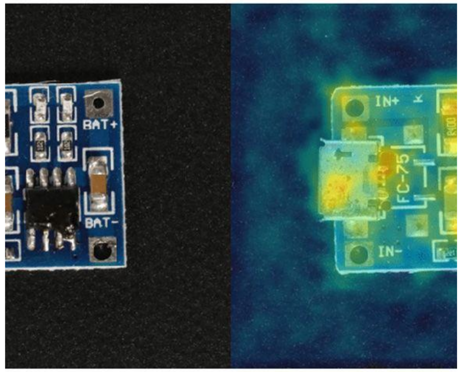
Agenda

Imaging

Camera Pipeline

Display

Image Quality & Enhancement



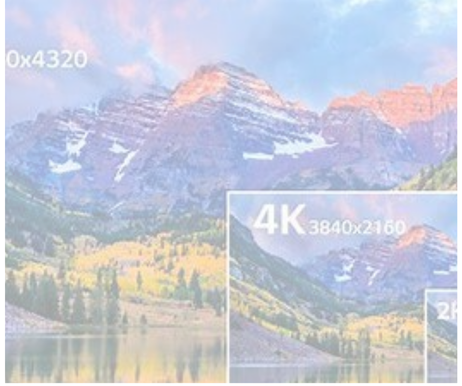
Manufacturing

Visual Inspection

Agenda

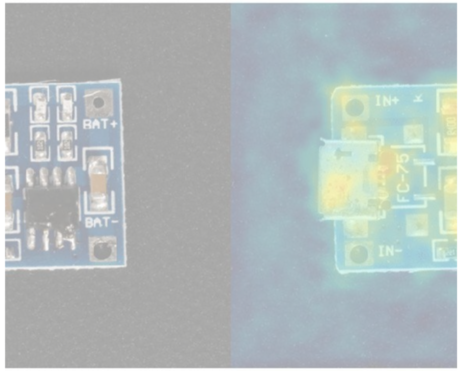
Imaging

Camera Pipeline



Display

Image Quality & Enhancement

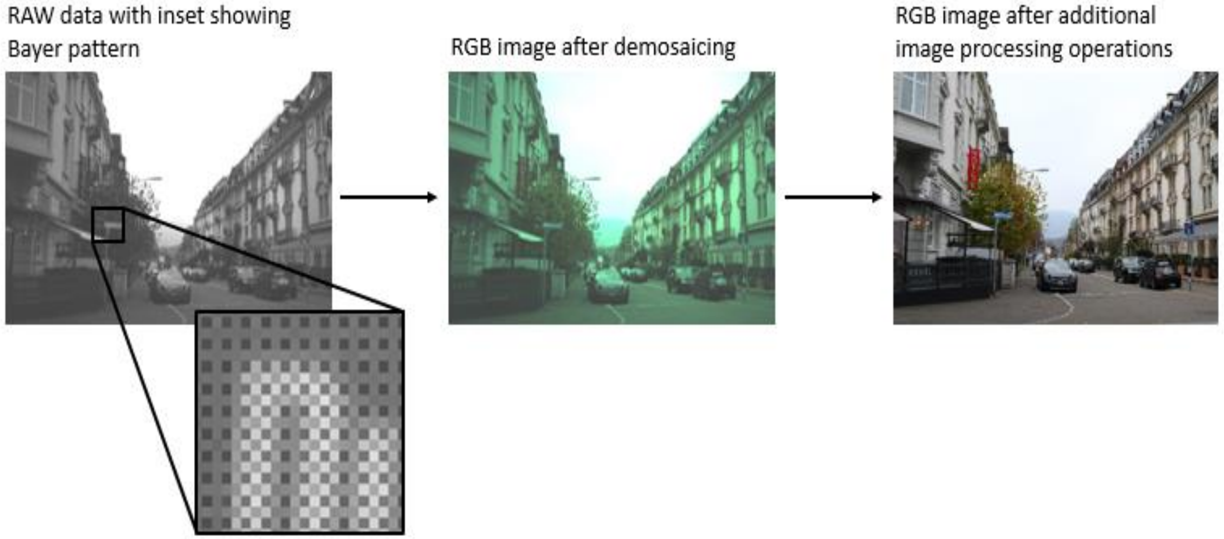


Manufacturing

Visual Inspection

Camera Pipeline Design – Traditional and Deep Learning

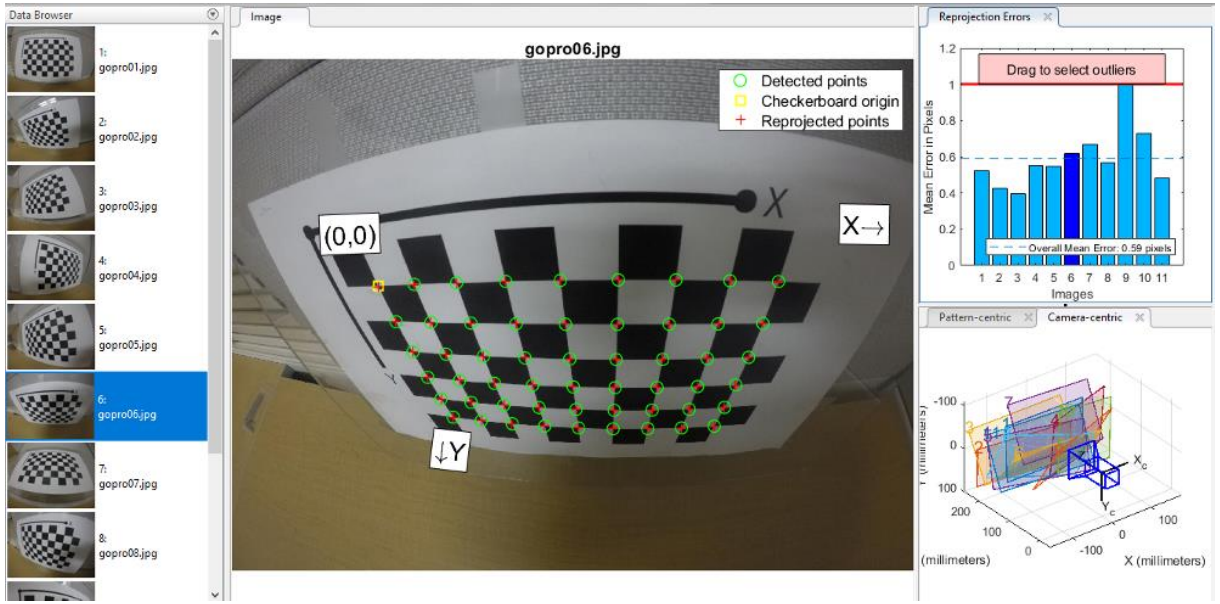
Implement Digital Camera Processing Pipeline

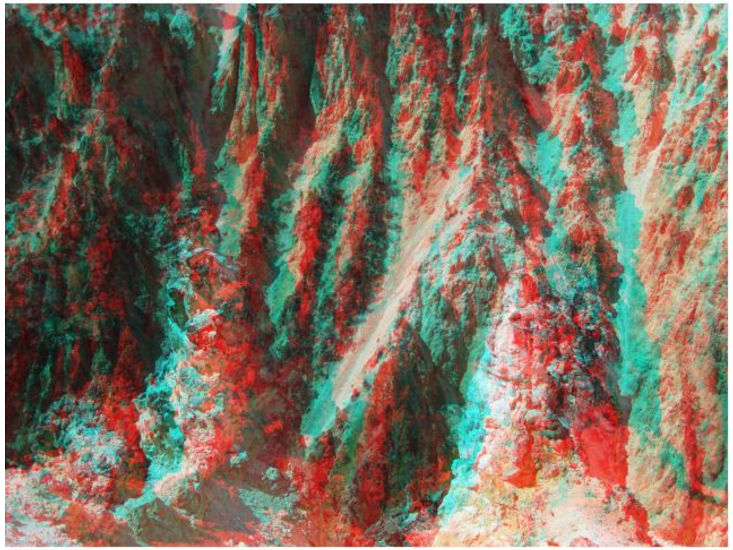


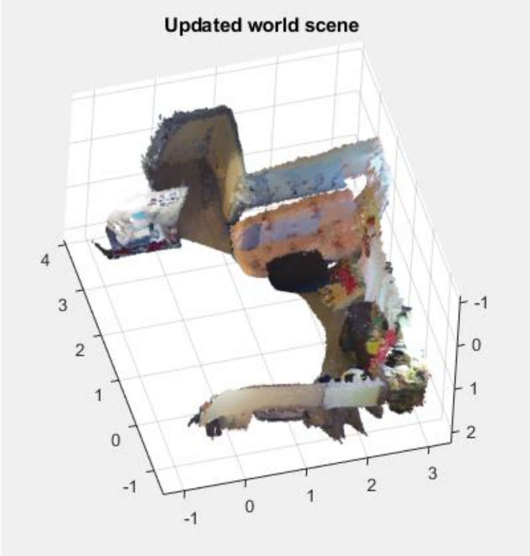
Develop Camera Processing Pipeline Using Deep Learning

Camera Calibration

- Estimate camera intrinsic and extrinsic parameters (including fisheye)
- Remove the effects of lens distortion
- Measure sizes of real-world objects
- Compute stereo disparity and depth
- Structure from motion



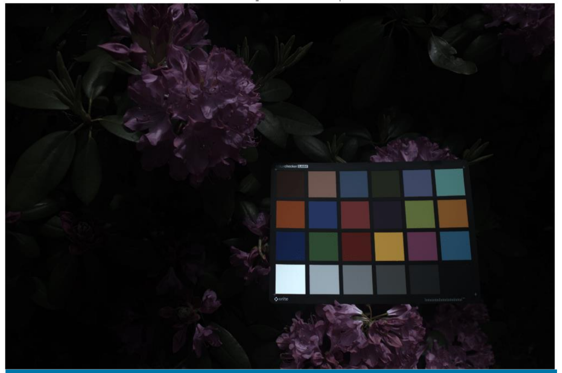




RAW to RGB Camera Pipeline

- Import RAW files formats such as Nikon NEF, Canon CRW and Adobe DNG, and read CFA image
- Linearize CFA image data
- Scale the images and apply white-balance adjustment
- Demosaic and Rotate image
- Convert CFA Image to RGB image (or sRGB)

Linear CFA Image



Rendered Image in Linear Color Space

Functions	
rawinfo	Read information about color filter array (CFA) images in RAW files
rawread	Read CFA images from RAW files
demosaic	Convert Bayer pattern encoded image to truecolor image
lin2rgb	Apply gamma correction to convert linear sRGB to sRGB color space
raw2planar	Separate a Bayer-patterned CFA image into individual, sensor-element
planar2raw	Combine planar sensor images into a full Bayer-pattern CFA image
raw2rgb	**Convert a RAW file into an RGB file in one step**

Rendered RGB Image in sRGB Color Space

Example: End-to-End Implementation of Digital Camera Processing Pipeline

White Balance Algorithms

Automatic white balancing is done in two steps:

- Step 1: Estimate the scene illuminant
- Step 2: Correct the color balance of the image

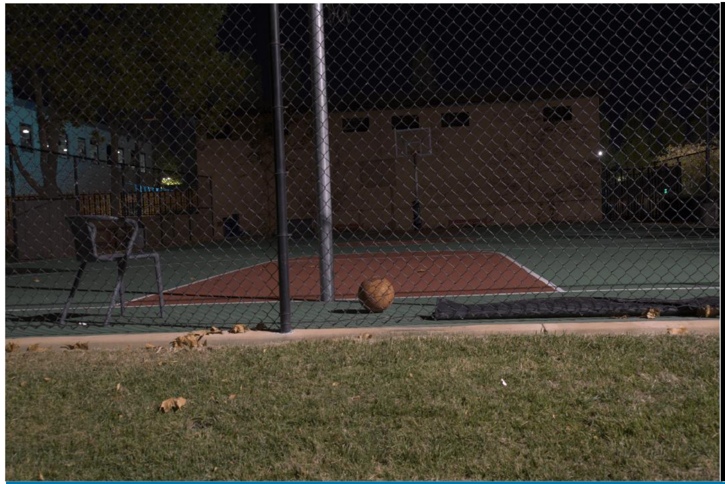
Algorithms to estimate scene illuminant	
illumwhite	Estimate illuminant using White Patch Retinex algorithm
illumgray	Estimate illuminant using gray world algorithm
illumpca	Estimate illuminant using principal component analysis (PCA)

Montage of Best White-Balanced Images: White Point, Gray World, Cheng

Example: Comparison of Auto White Balance Algorithms

Recover Low Light Images Using Deep Learning

- Recover RAW images taken in low light with short exposure times using deep learning network
- Pretrained low-light recovery U-Net deep learning network



Low Light Long Exposure

Low Light Short Exposure

Low Light Short Exposure Recovered with Deep Learning

[Example: Brighten Extremely Dark Images Using Deep Learning]

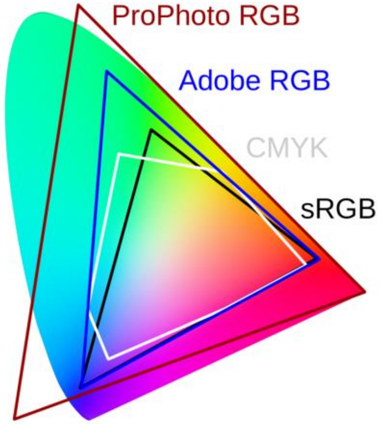
Conversion Between Color Spaces

Support for wide-gamut color spaces

- BT.2020 (Ultra High Definition, UHD)
- BT.2100 (High Dynamic Range, HDR)
- ProPhoto (ROMM RGB) color space

4K HDR, Source: Pexels

Function	
rgbwide2ycbcr	Convert wide-gamut RGB color values to YCbCr color values
ycbcr2rgbwide	Convert YCbCr color values to wide-gamut RGB color values
xyz2rgbwide	Convert CIE 1931 XYZ color values to wide-gamut RGB color values
rgbwide2xyz	Convert wide-gamut RGB color values to CIE 1931 XYZ color values



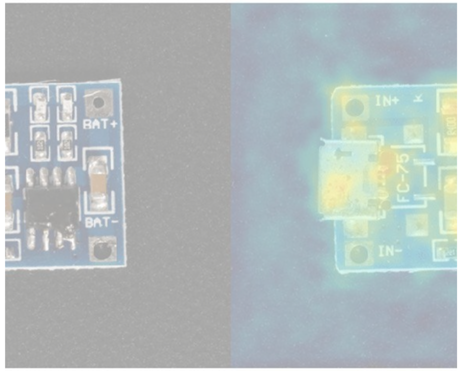
Agenda

Imaging

Camera Pipeline

Display

Image Quality & Enhancement



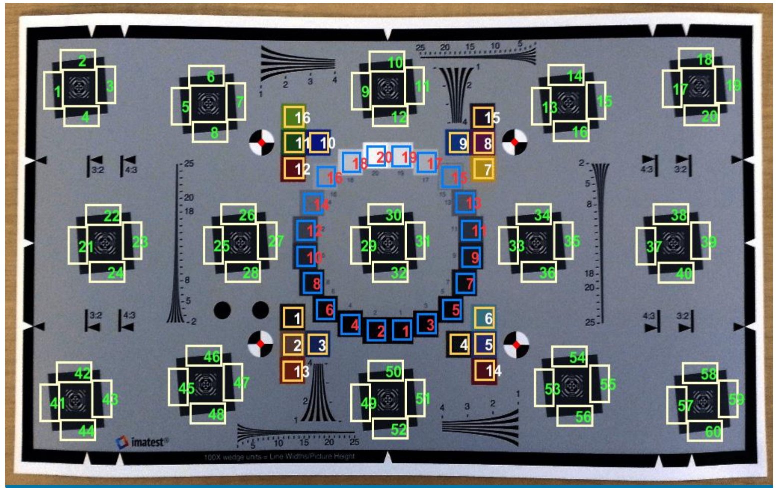
Manufacturing

Visual Inspection

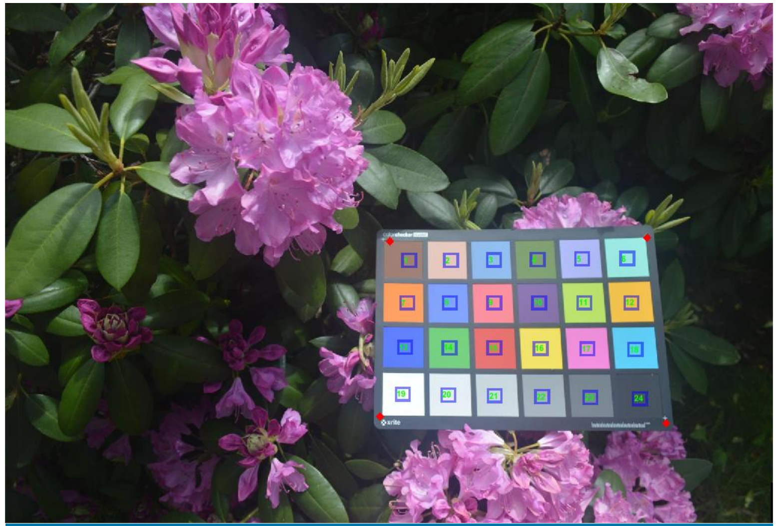
Test Charts Support

- Detection and analysis of enhanced, wedge enhanced, and wedge extended versions of Imatest eSFR test charts (ISO 12233)
- Support for X-Rite® (Gretag Macbeth®) ColorChecker ® test charts

Function	
measureIlluminant	Measure scene illuminant of test chart
colorChecker	Identifies the color patch ROI in test chart
displayChart	Display test chart with ROI
measureColor	Measure colors in test chart
displayColorPatch	Display measured and reference color as color patches
plotChromaticity	



Enhanced, Wedge Enhanced, and Wedge Extended eSFR test chart



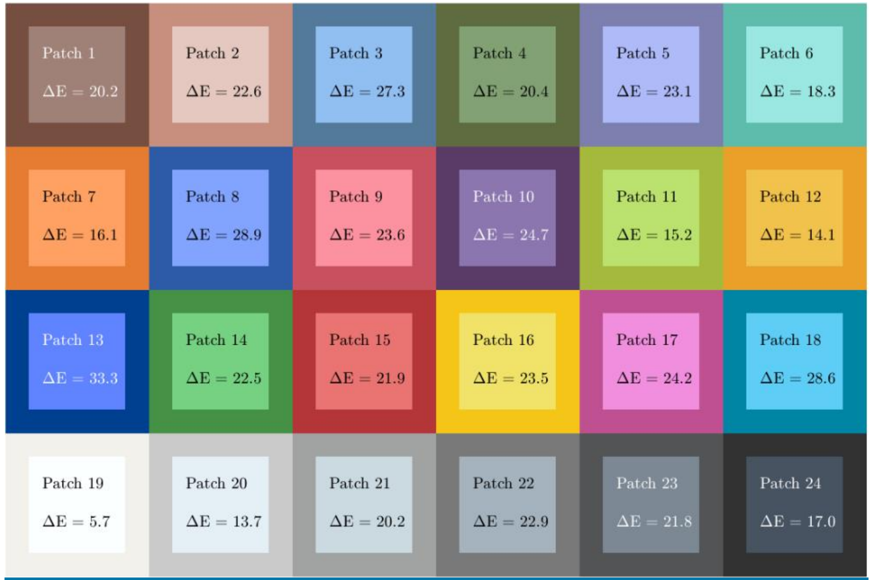
X-Rite ColorChecker

Example: Evaluate Quality Metrics on eSFR Test Chart   Example: colorChecker

Quality Measurement by Image Comparison

- Measure color deviations in test charts
- Compare color difference between 2 images in RGB or L*a*b color space

Function	
deltaE	Color difference based on CIE76 standard
imcolordiff	Color difference based on CIE94/CIE2000 standard

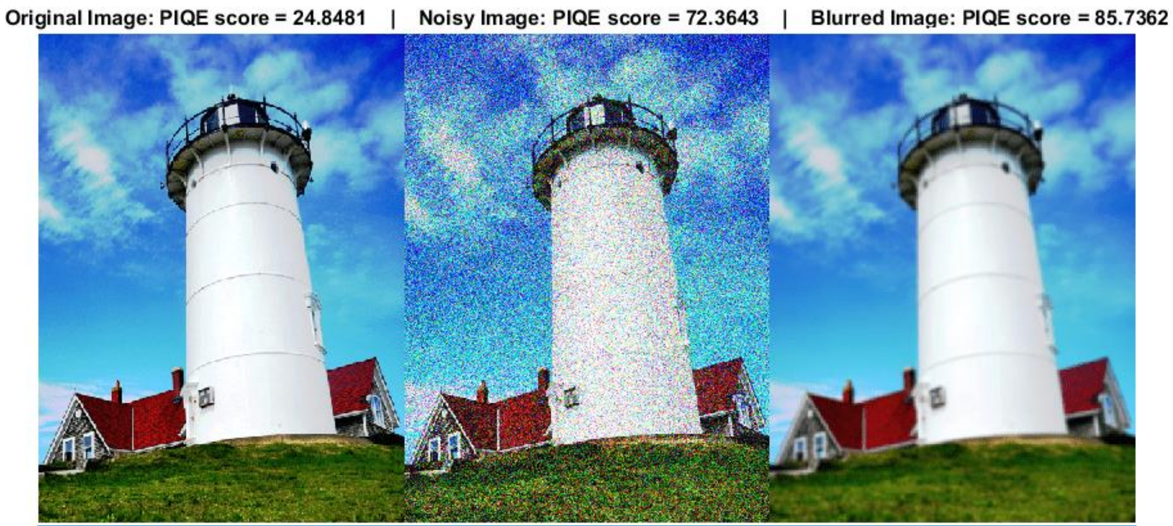


Measured vs Reference Color Difference

Original vs Local Color Difference

Image Quality Metrics

Full Reference Techniques Function	
immse	Mean squared error
psnr	Peak signal-to-noise
ssim	Structural similarity metric
multissim	MS-SSIM index for image quality
multissim3	MS-SSIM index for volume quality
No-Reference Techniques Function	
niqe	Naturalness Image Quality Evaluator
brisque	Blind Reference-less Image Spatial Quality Evaluator
piqe	Perception-based Image Quality Evaluator



PIQE No-Reference Techniques

Example: Train and Use a No-Reference Quality Assessment Model

Enhance Low Light Image using Dehazing Algorithm

Using haze removal techniques to enhance low-light images comprises three steps:

- Step 1: Invert the low-light image
- Step 2: Apply the haze removal algorithm to the inverted low-light image
- Step 3: Invert the enhanced image

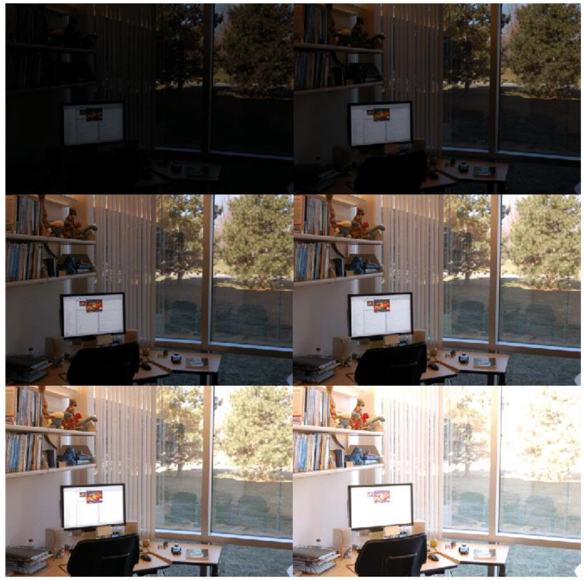
[Example: Low-Light Image Enhancement]

Neural Style Transfer Using Deep Learning

Example: Train Fast Style Transfer Network

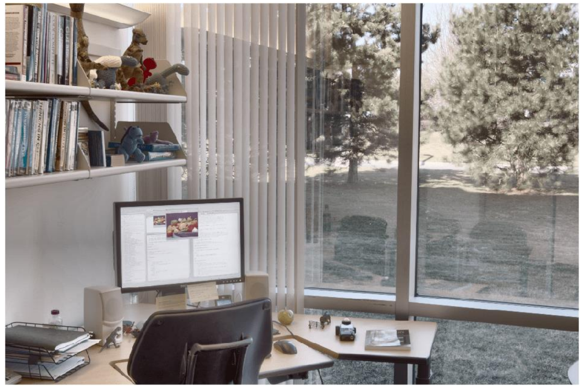
Example: Neural Style Transfer Using Deep Learning

High Dynamic Range (HDR) Images



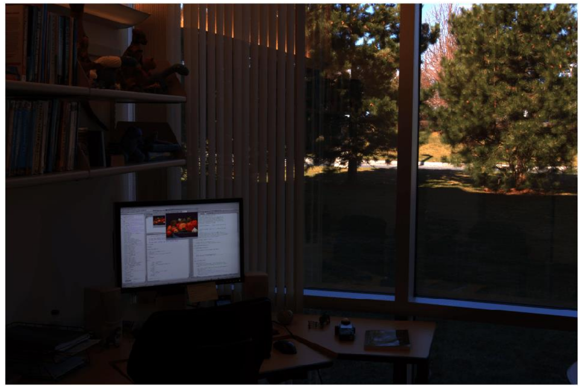
makehdr

Create the HDR from the set of LDR



tonemap

Convert HDR to LDR



Support EXR and HDR files

Increase Image Resolution

Set of Low-Resolution Burst Mode Images

High-Resolution Image

Create high-resolution image from <u>set of low-resolution burst mode images</u>

High-Resolution Results Using Bicubic Interpolation (Left) vs. VDSR (Right)

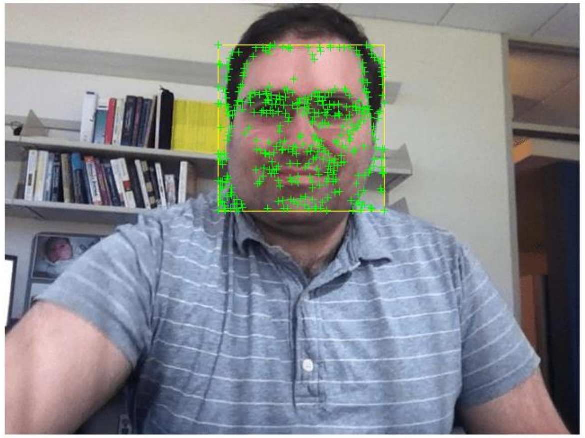
<u>Single image super-resolution (SISR)</u> using a very-deep super-resolution (VDSR) neural network

Face Detection and Tracking

- Figure out where people are to make localized adjustments (accentuate the person and minimize the background, auto-focus assistance)
- Develop the system in three steps:
  - Step 1: Detect a face
  - Step 2: Identify facial features to track
  - Step 3: Track the face

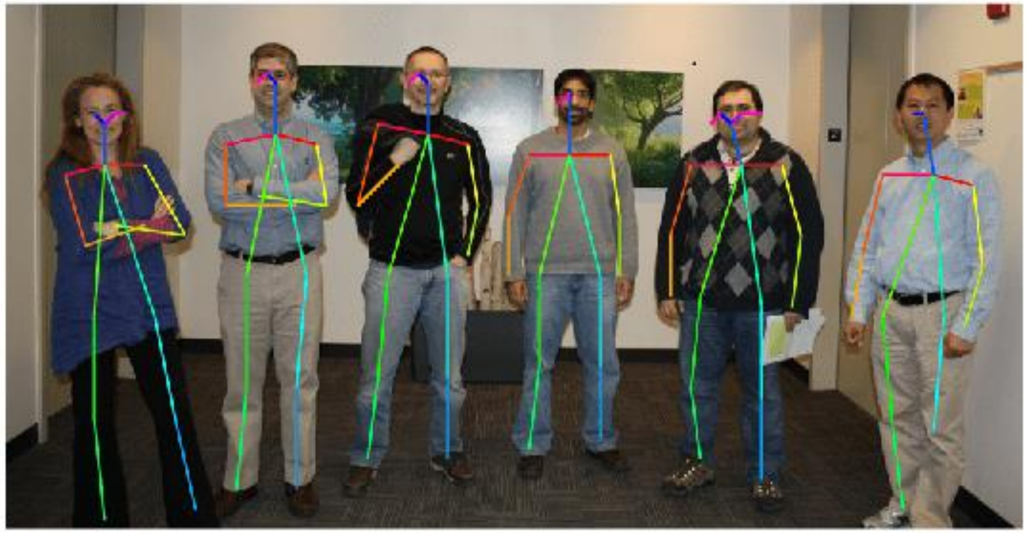
Detected face

Detected features



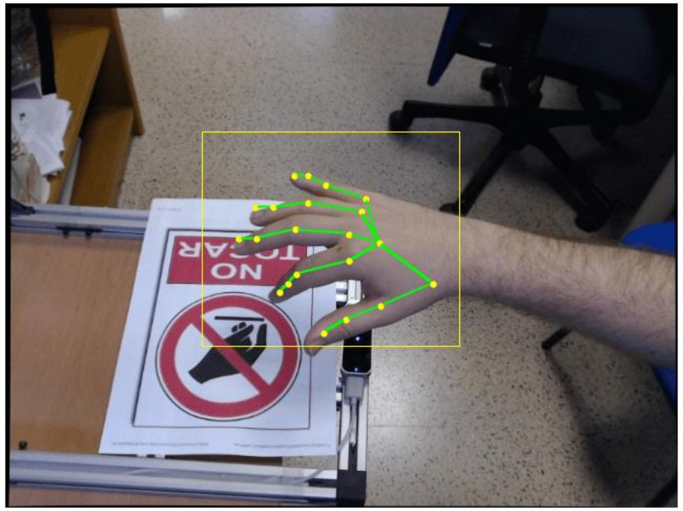
Pose Estimation

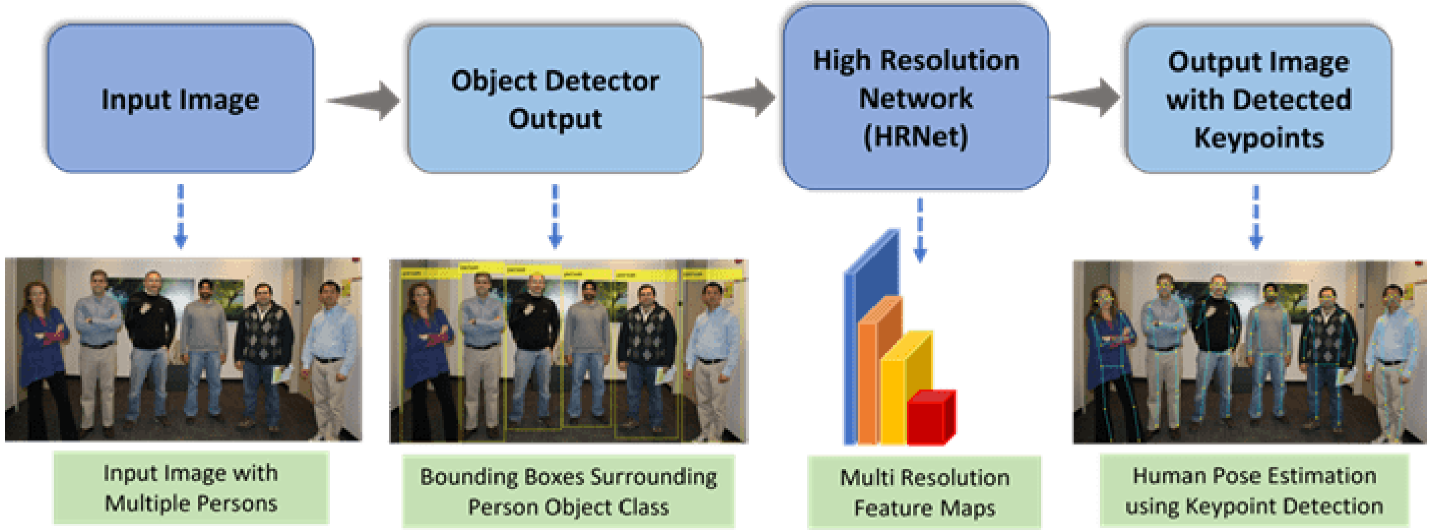
- Identify the location of people in an image and the orientation of their body parts.
- OpenPose is a multi-person human pose estimation algorithm that uses a bottom-up strategy.
- A bottom-up strategy first identifies body parts in an image, such as noses and left elbows, and then assembles individuals based on likely pairings of body parts.

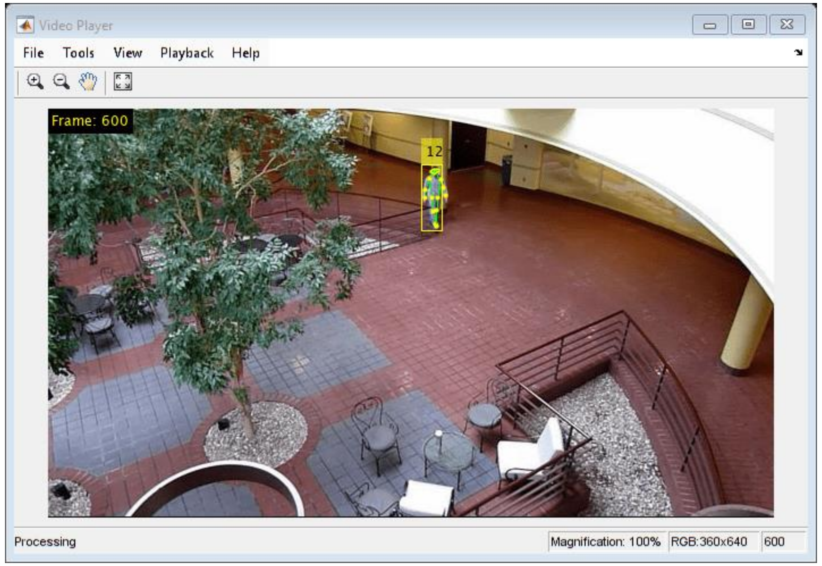


Multi-Object Tracking and Human/Hand Pose Estimation

- **Detect** people in each video frame using a YOLO v4
- **Track** the detected people across frames using trackerGNN uses linear Kalman filter
- **Identify** keypoints and estimate body/hand poses using HRNet



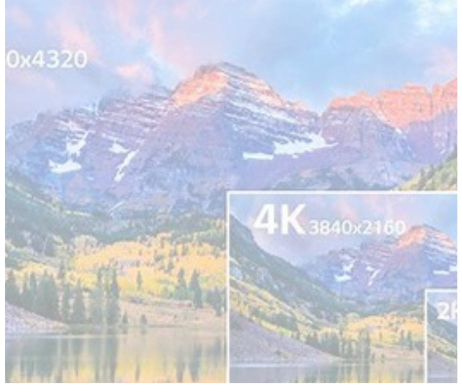




Agenda

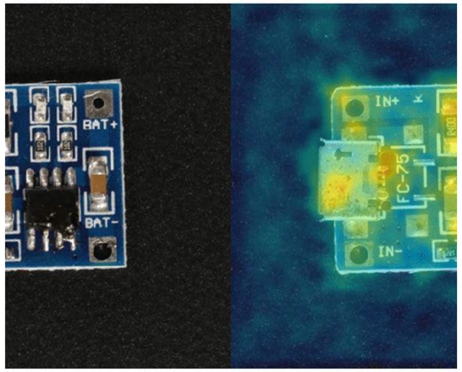
Imaging

Camera Pipeline



Display

Image Quality & Enhancement



Manufacturing

Visual Inspection

What is Automated Visual Inspection?

*" Automated optical inspection is the **image-based** or **visual inspection** of manufacturing parts where a camera scans the device under test for both **failures** and **quality defects**"*

Automated Defect Detection

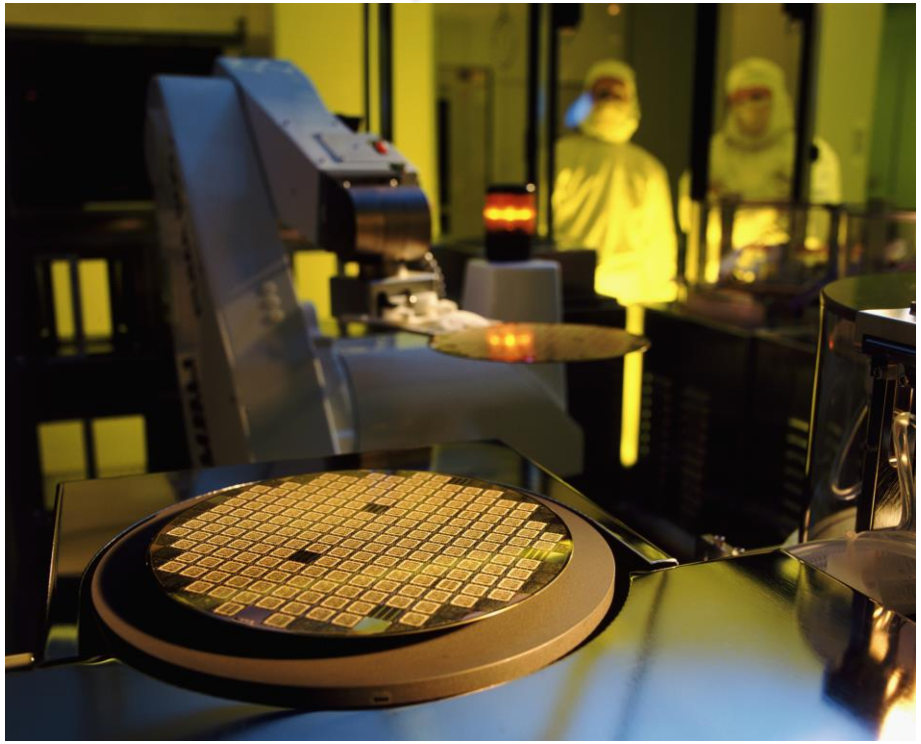
Machine Vision

Optical Inspection

Automated Inspection

Typical Visual Inspection System

Inspection Cameras



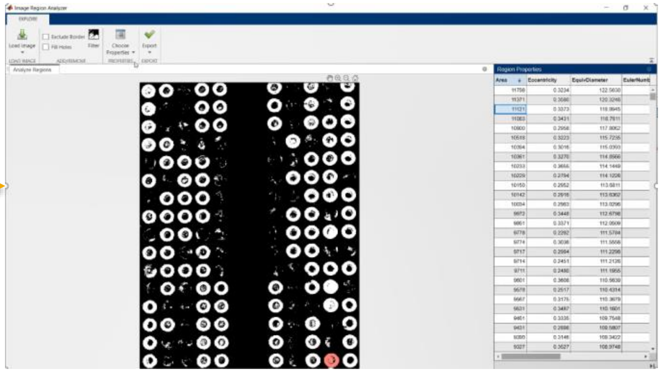
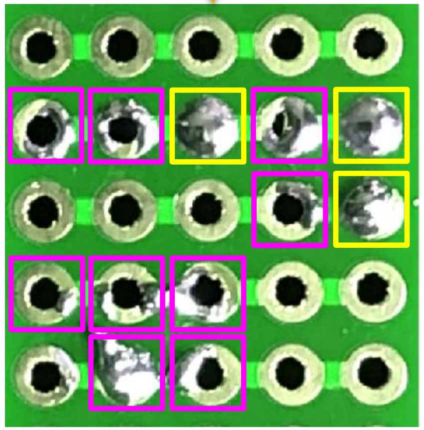


Image Analysis



Defective Parts

AI-based Visual Inspection Workflow

Data Preparation

- Data cleansing and preparation
- Human insight
- Simulation-generated data

AI Modeling

- Model design and tuning
- Hardware accelerated training
- Interoperability

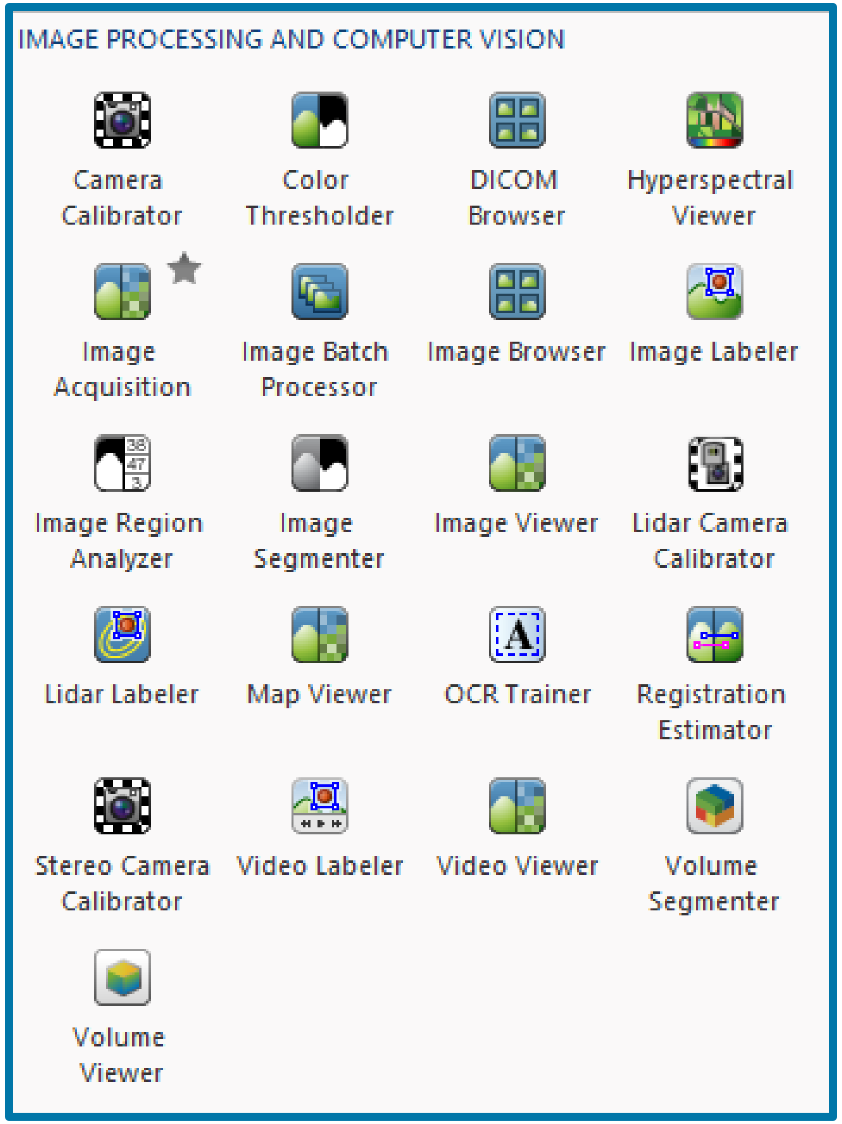
Simulation & Test

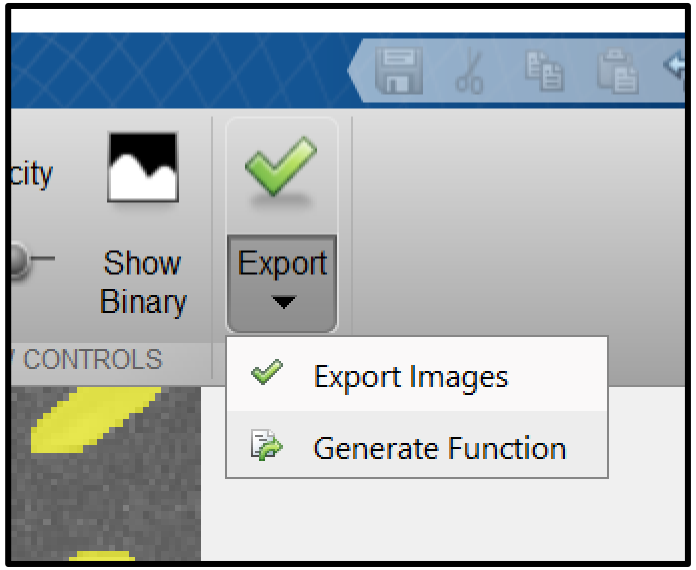
- Integration with complex systems
- System simulation
- System verification and validation

Deployment

- Embedded devices
- Enterprise systems
- Edge, cloud, desktop

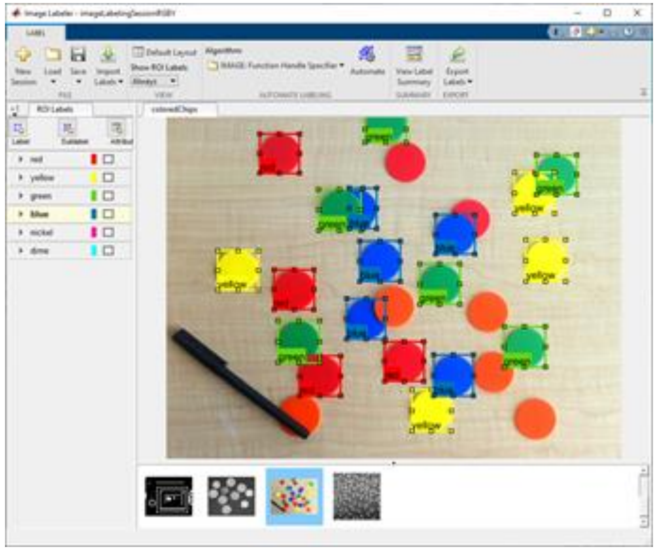
Apps Accelerate Workflow



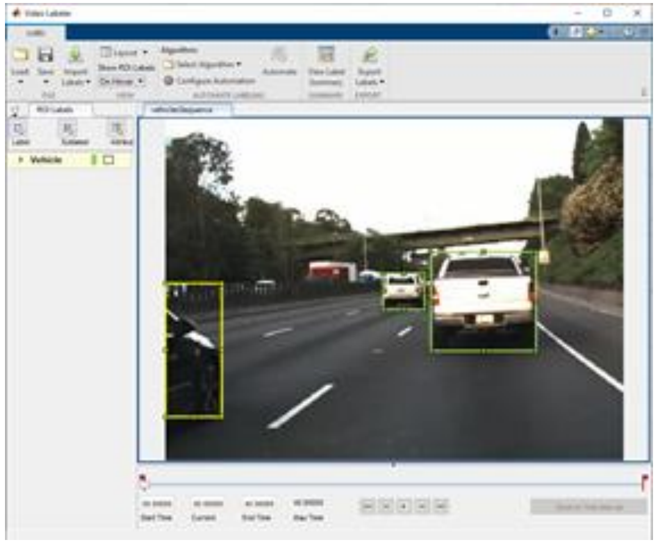


Data Labeling

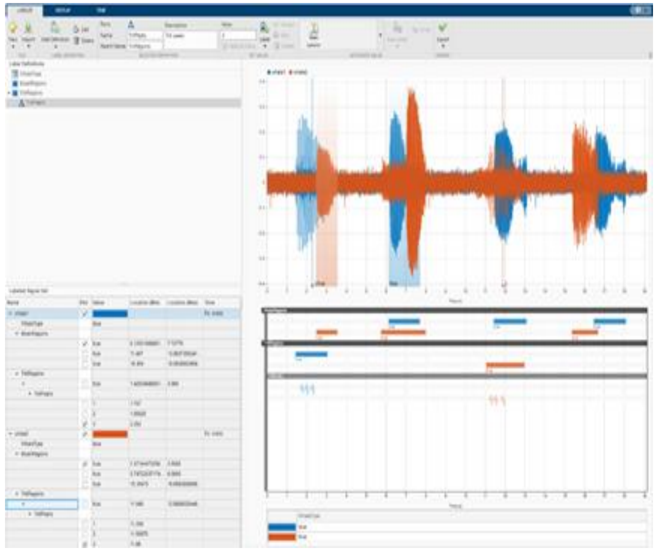
Image Labeler



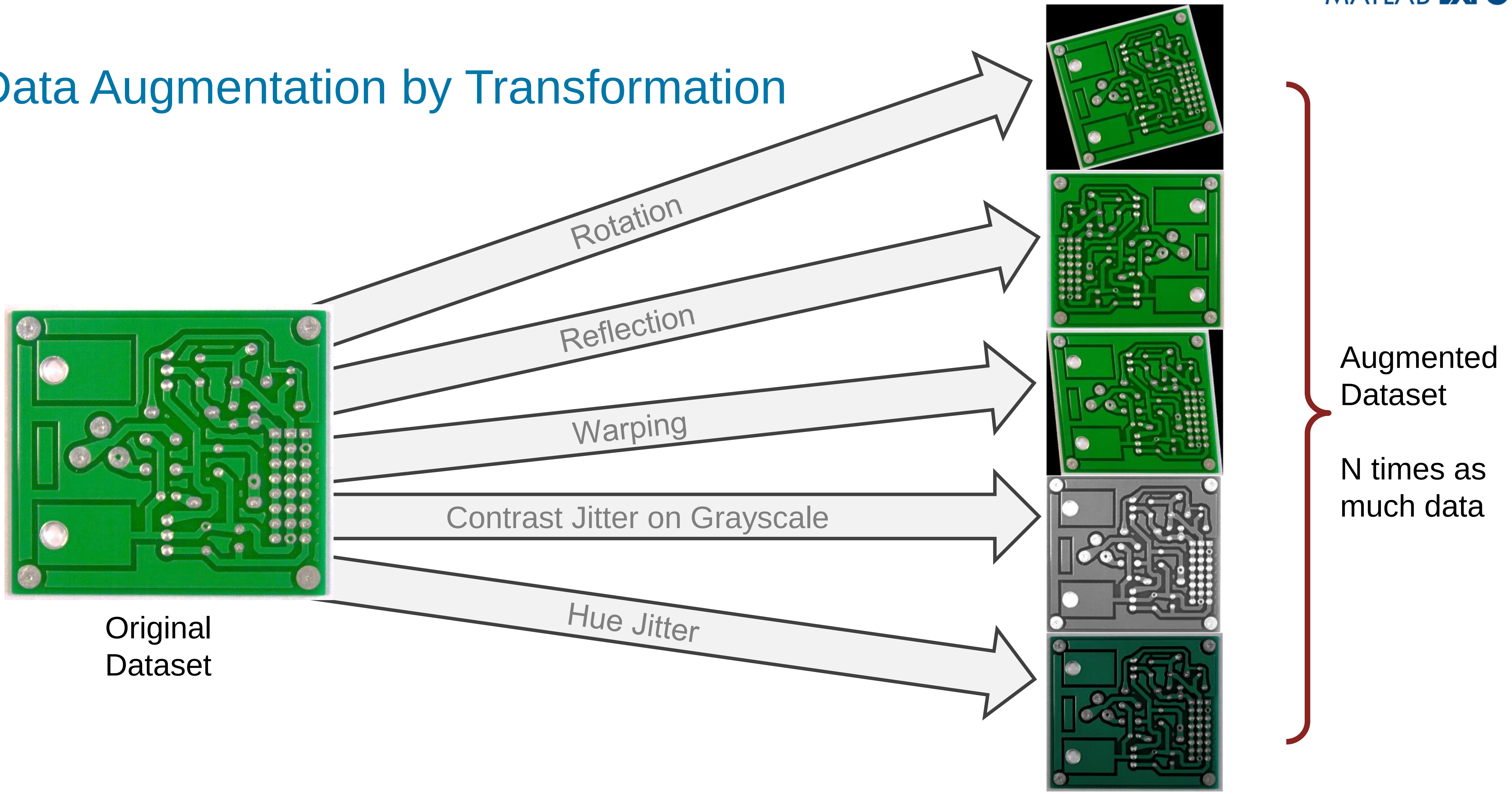
Video Labeler



Signal Labeler



Data Augmentation by Transformation



Data augmentation allows building more complex and more robust models

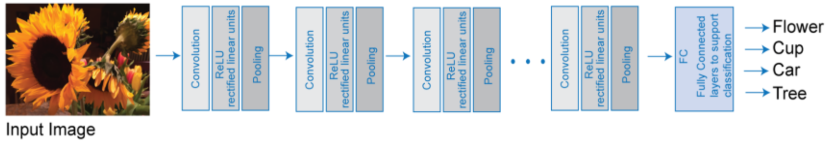
Data Augmentation using Generative Adversarial Networks (GANs)

Images of digits generated from noise.

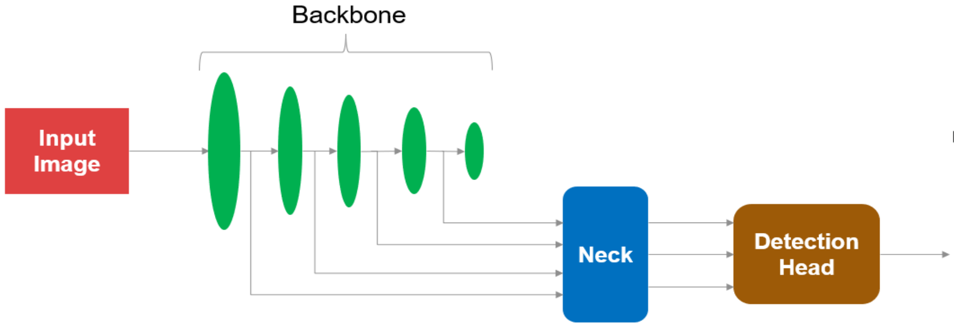
File Exchange: Conditional GAN (Generative Adversarial Network) with MNIST

Architectures for Visual Inspection

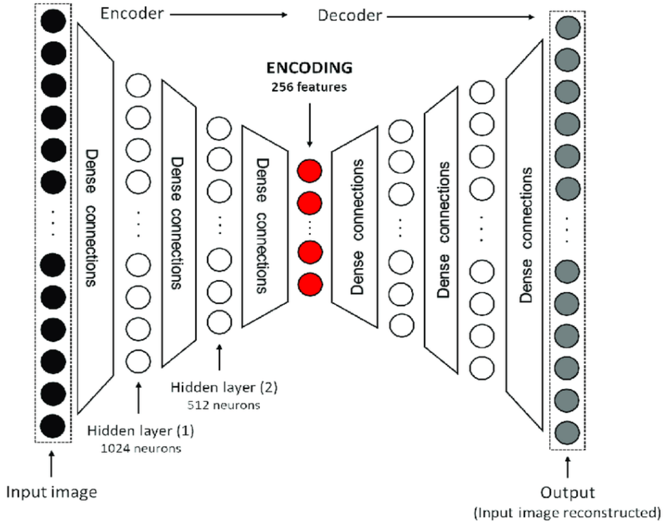
Convolutional Neural Networks



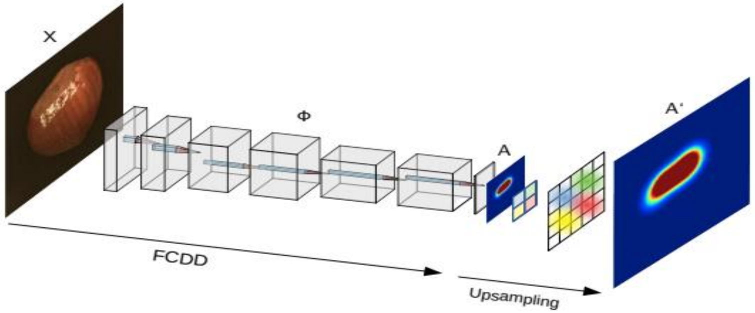
YOLO – Object Detector



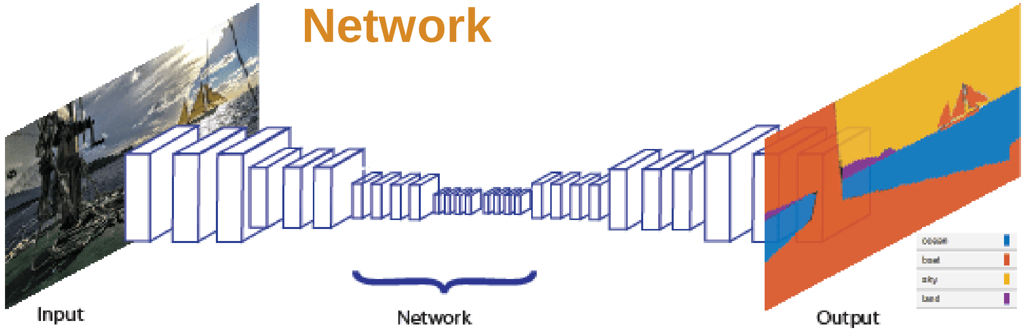
Deep Autoencoder



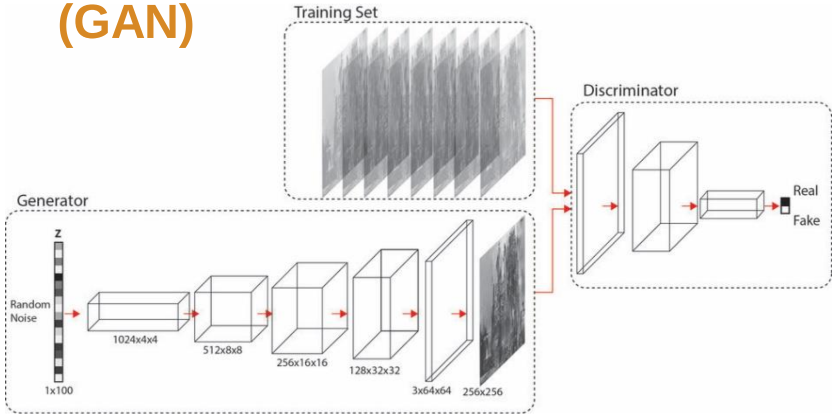
Fully Convolutional Data Description (FCDD) – Anomaly Detector



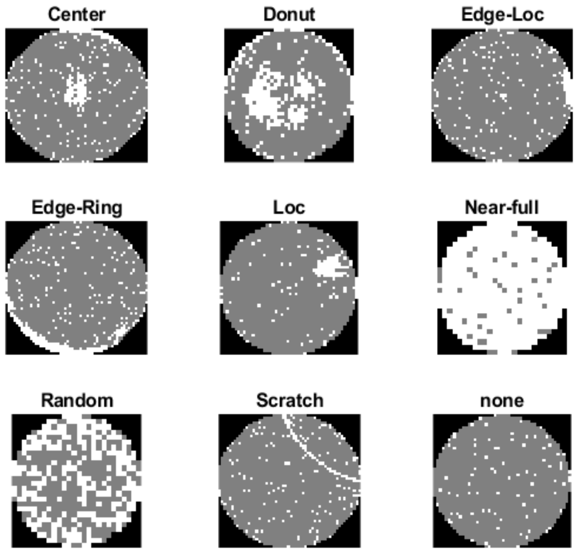
Semantic Segmentation Network



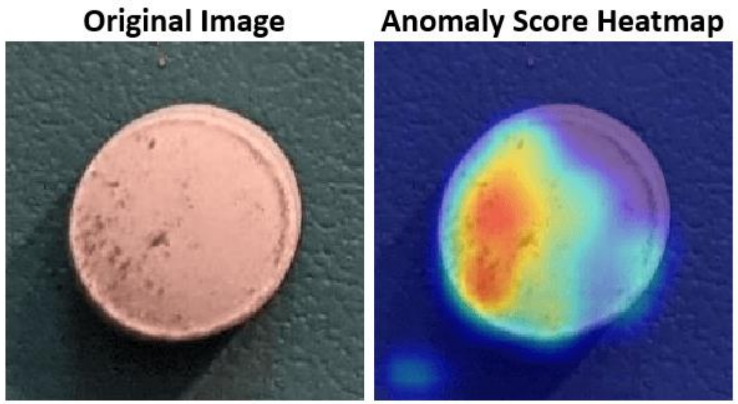
Generative Adversarial Network (GAN)



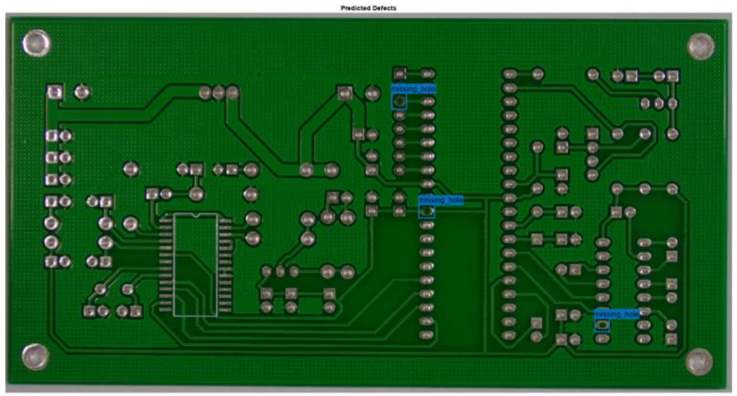
Visual Inspection Examples



Defect Classification



Anomaly Detection

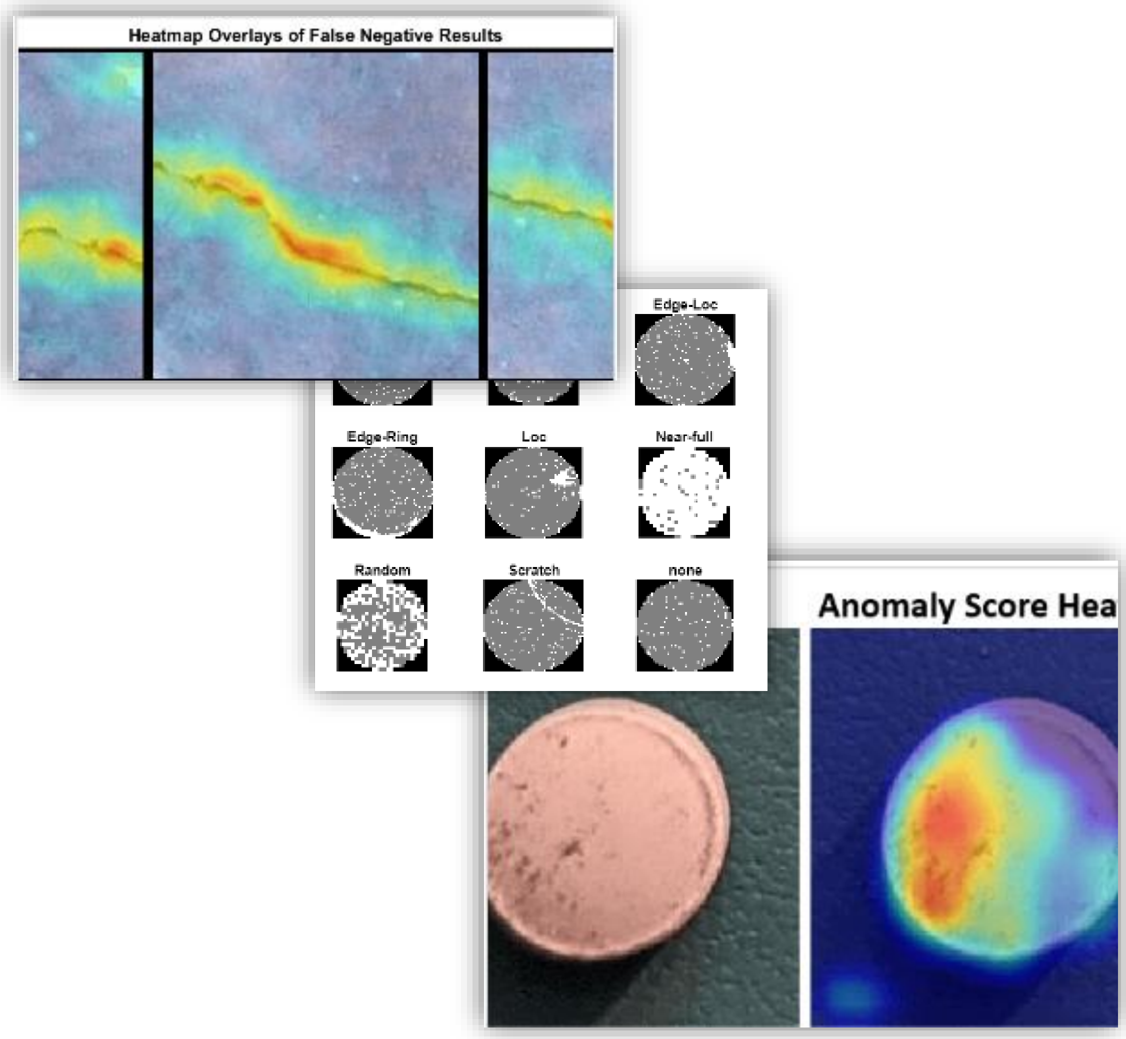


Localization and Classification on PCBs

Visual Inspection Support package

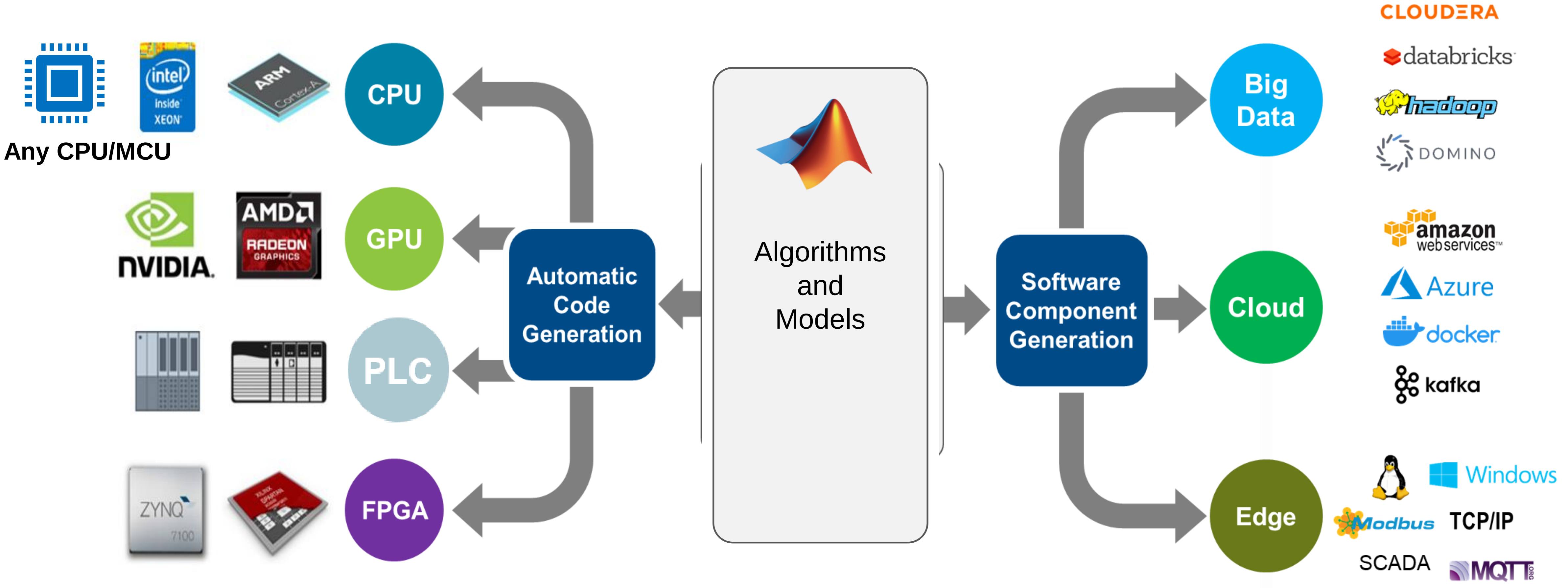
[Computer Vision Toolbox Automated Visual Inspection Library]

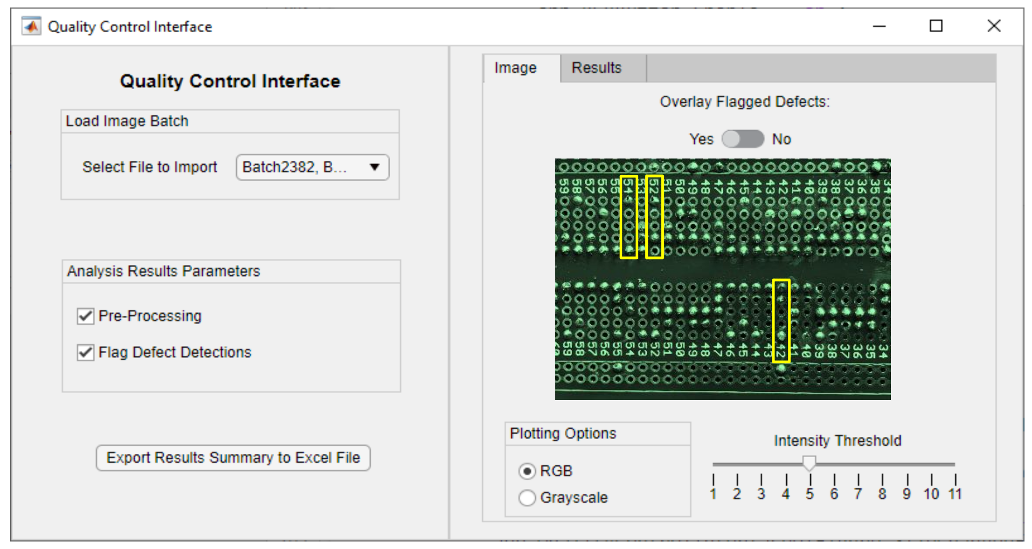
- Anomaly detector
- Parameter optimization
- Visualization and evaluation tools
- Dedicated examples
  - Detect Image Anomalies Using Pretrained ResNet-18 Feature Embeddings
  - Classify Defects on Wafer Maps Using Deep Learning
  - Detect Image Anomalies Using Explainable One-Class Classification Neural Network



Deploy to Enterprise Infrastructure or Embedded Systems

AI models in MATLAB and Simulink can be deployed on enterprise systems or the cloud, or on embedded devices.

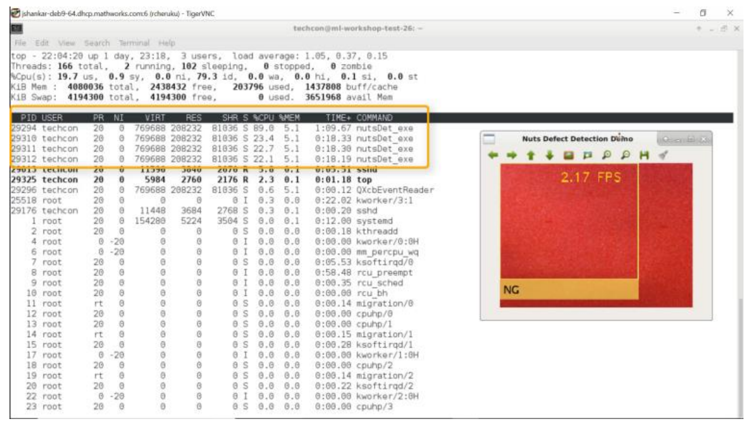




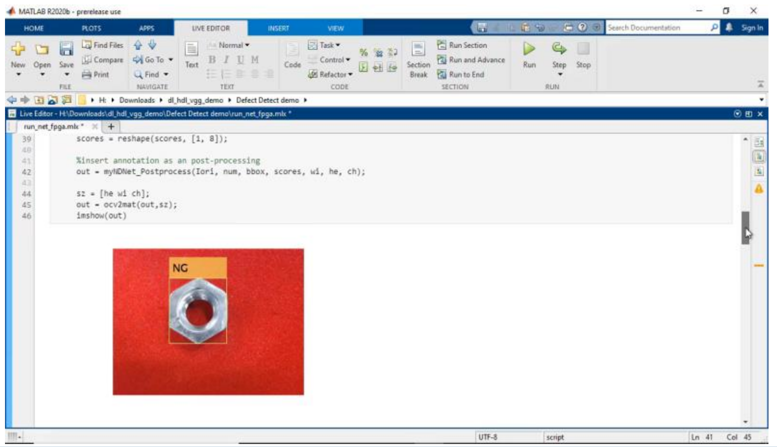
Deploy as Desktop or Web App

Deploy to Jetson AGX Xavier

Deploy to ARM Cortex-A microprocessor



Deploy to ZCU102 board from Xilinx



Wuhan JINGCE Electronic: Rapid Development of Model-Based Display Measuring Instruments

Challenge

Rapidly develop high-precision measurement instruments for display devices based on complex image processing and deep learning, measure and debug a series of optical parameters such as flicker, chromaticity and color uniformity, and spectrum.

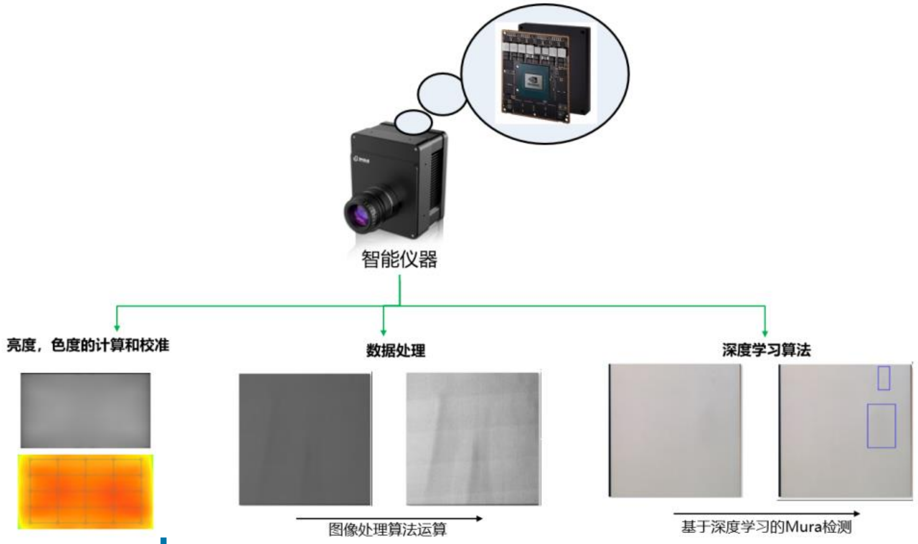
Solution

Use MATLAB to develop image processing algorithms, create and train neural networks, and use GPU Coder for automatic code generation to quickly implement productization.

Result

- The automatically generated CUDA code is 23.6% faster than the manually optimized code
- From algorithm prototype to product prototype, the development time is shortened by 35%~50%
- Product Engineering Quality Assurance (Gold Reference, SIL, PIL)
- Simplified processes and smaller teams, 30% reduction

精测电子用于显示屏高精量测的谱系化产品



量测一体化智能仪器

[>> Link to the speech at MATLAB EXPO China](#)

MATLAB EXPO

Thank you

MathWorks®

© 2024 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See *mathworks.com/trademarks* for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.