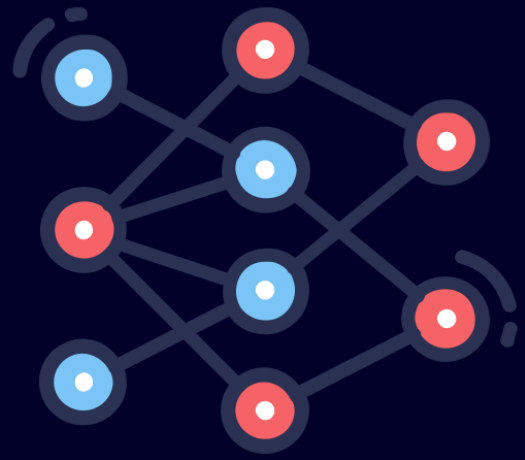
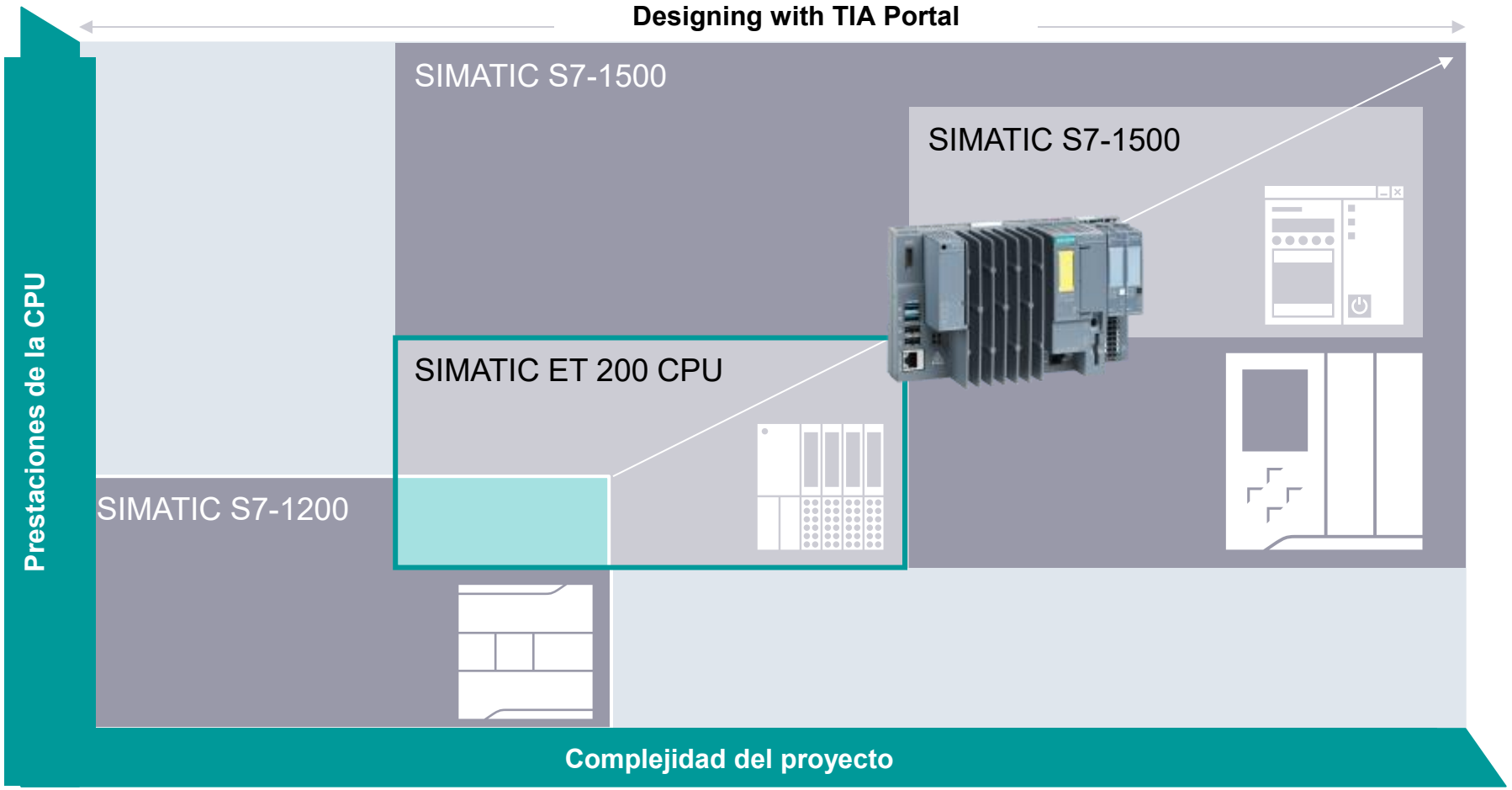
Redes neuronales en controladores SIMATIC

Congreso AADECA Agosto 2025



SIEMENS

SIMATIC ET 200SP Open Controller



Effective design

Innovative design

Integrated Diagnostics

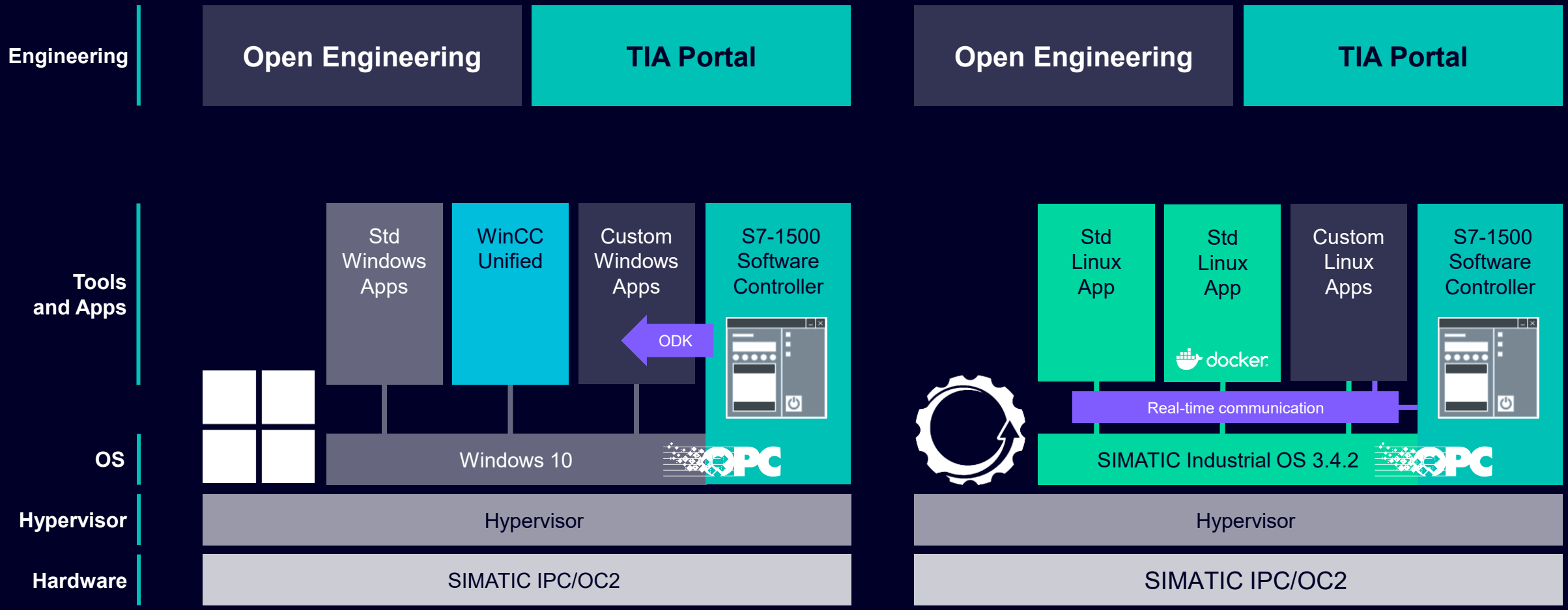
Safety Integrated

Security Integrated

Technology Integrated

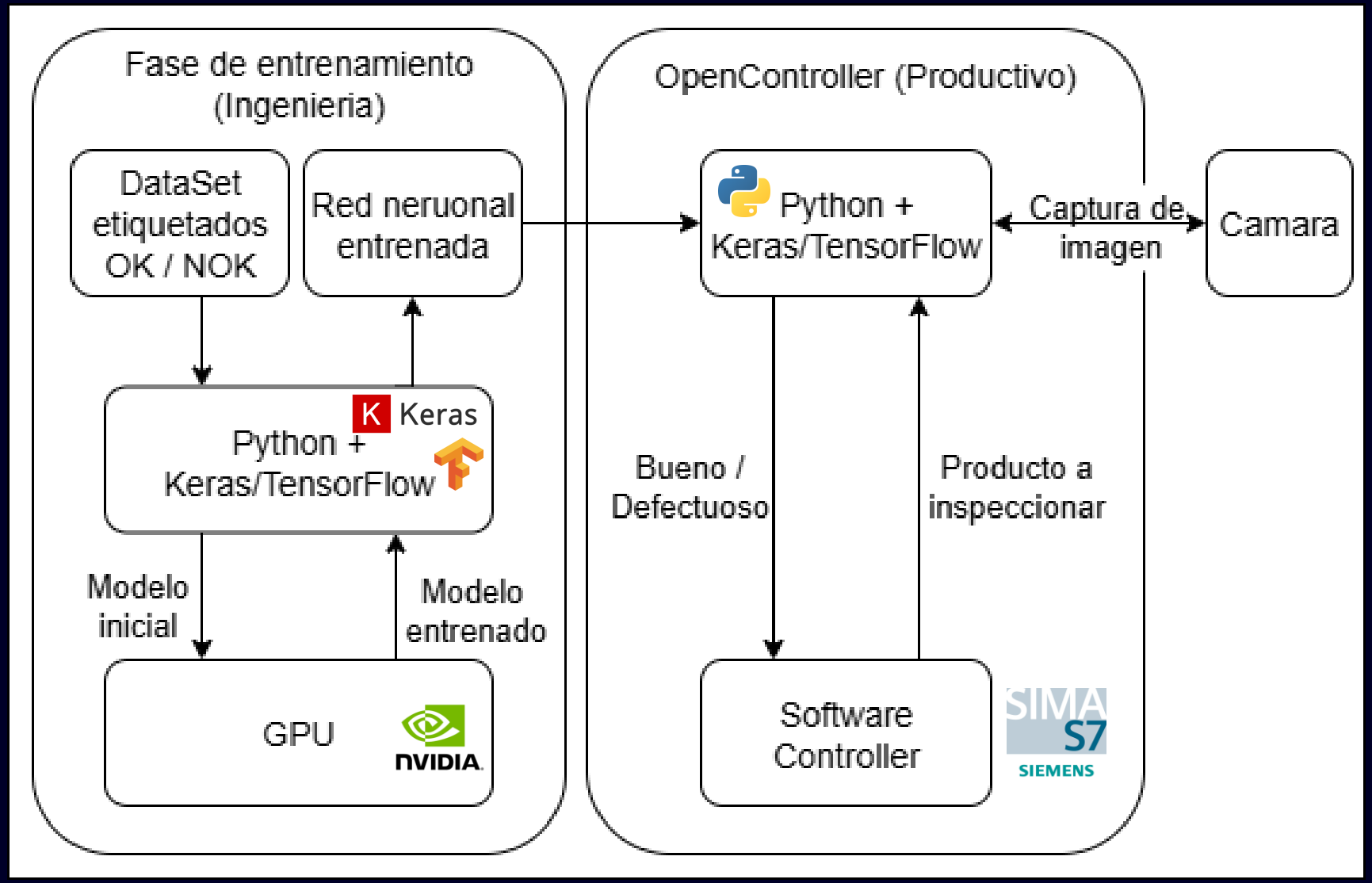
SIEMENS

SIMATIC S7-1500 Software Controller
Supported General Purpose OS Combinations



SIEMENS

Flujo de trabajo



SIEMENS

DataSet etiquetados

Buenas

Malas

SIEMENS

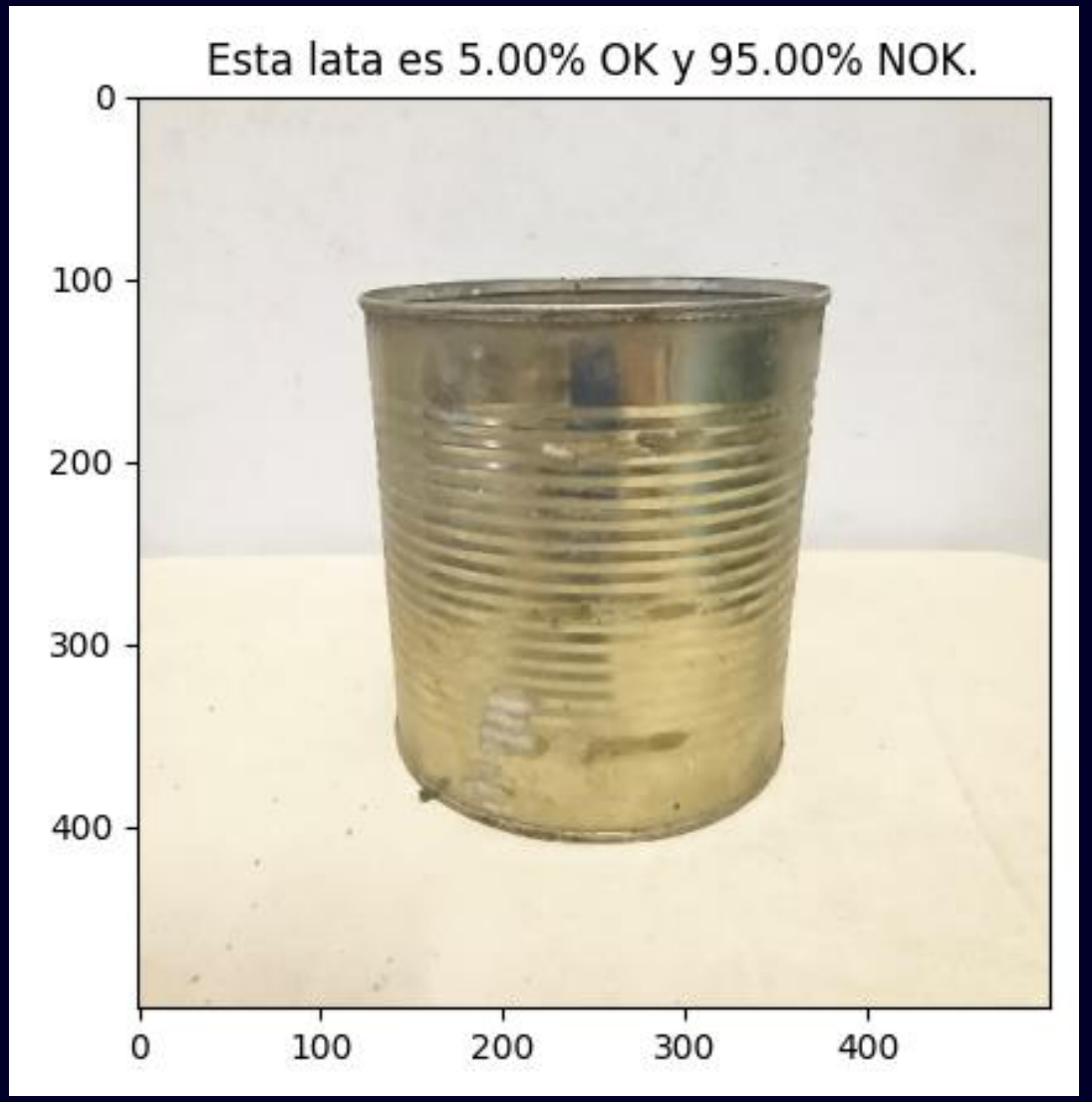
Programa de entrenamiento y ejecucion

```python
def make_model(input_shape, num_classes):
    inputs = keras.Input(shape=input_shape)
    x = data_augmentation(inputs)
    x = layers.Rescaling(1./255)(x)
    ....


    # Entry block
    x = layers.Rescaling(1.0 / 255)(inputs)
    x = layers.Conv2D(128, 3, strides=2, padding="same")(x)
    x = layers.BatchNormalization()(x)
    x = layers.Activation("relu")(x)

    previous_block_activation = x  # Set aside residual

    for size in [256, 512, 728]:
        x = layers.Activation("relu")(x)
        x = layers.SeparableConv2D(size, 3, padding="same")(x)
        x = layers.BatchNormalization()(x)

        x = layers.Activation("relu")(x)
        x = layers.SeparableConv2D(size, 3, padding="same")(x)
        x = layers.BatchNormalization()(x)

        x = layers.MaxPooling2D(3, strides=2, padding="same")(x)

        # Project residual
        residual = layers.Conv2D(size, 1, strides=2, padding="same")(
            previous_block_activation
        )
        x = layers.add([x, residual])  # Add back residual
        previous_block_activation = x  # Set aside next residual

    x = layers.SeparableConv2D(1024, 3, padding="same")(x)
    x = layers.BatchNormalization()(x)
    x = layers.Activation("relu")(x)

    x = layers.GlobalAveragePooling2D()(x)
    if num_classes == 2:
        units = 1
    else:
        units = num_classes

    x = layers.Dropout(0.25)(x)
    # We specify activation=None so as to return logits
    outputs = layers.Dense(units, activation=None)(x)
```

```python
import os
os.environ['TF_ENABLE_ONEDNN_OPTS'] = '0'
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'
import numpy as np
import keras
from keras import layers
from tensorflow import data as tf_data
import matplotlib.pyplot as plt

image_size = (500, 500)
batch_size = 128

from keras.models import load_model


model = load_model("save_at_999.keras")


img = keras.utils.load_img("Images/mala1.jpg", target_size=image_size)
img_array = keras.utils.img_to_array(img)
img_array = keras.ops.expand_dims(img_array, 0)  # Create batch axis
predictions = model.predict(img_array)
score = float(keras.ops.sigmoid(predictions[0][0]))
print(f"Img1 is {100 * (1 - score):.2f}% Buena and {100 * score:.2f}% Mala.")
text = f"This image is {100 * (1 - score):.2f}% Buena and {100 * score:.2f}% Mala."
plt.imshow(img)
plt.title(text)
plt.show()
```


SIEMENS

Resultado del modelo



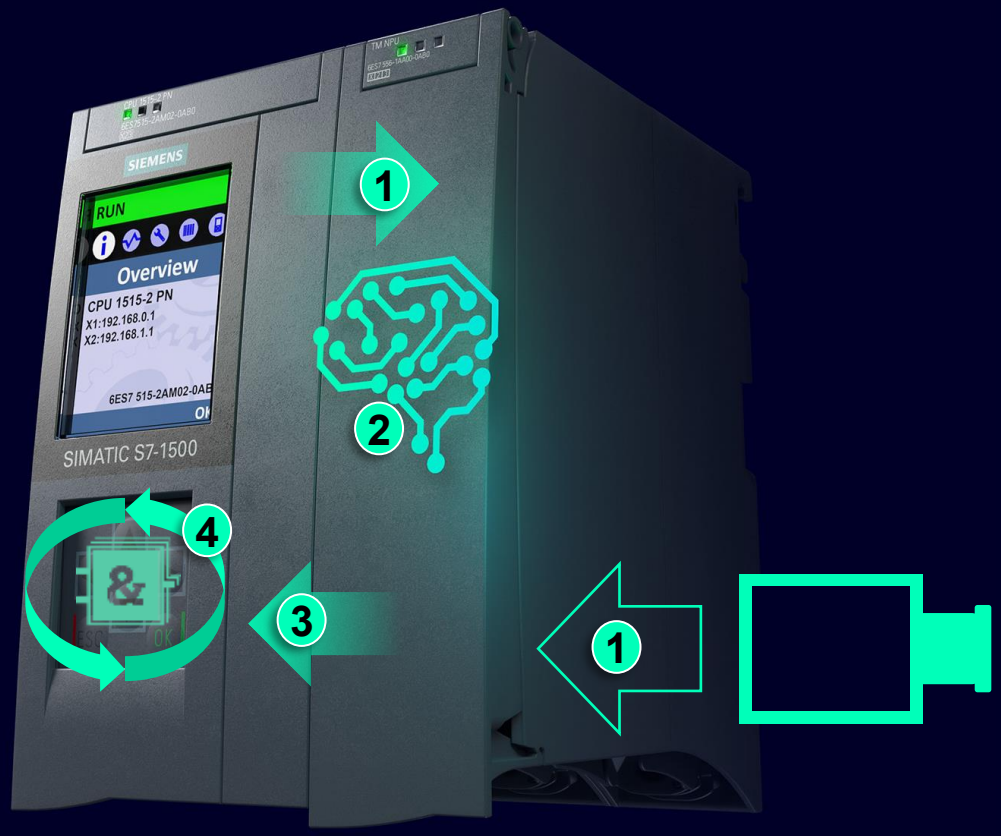
SIEMENS

Artificial Intelligence in SIMATIC
S7-1500 TM NPU (Neural Processing Unit)

+ SIMATIC S7-1500 Technology Module for use in central and distributed configurations
+ Engineering and handling via TIA Portal
+ Connection of Camera via Ethernet interface

TM NPU processing procedure:

1. Data of the PLC (sensors, actors, ...) is sent to the module / Connected Camera takes pictures of the production process
2. Processing of input data (images, PLC data) by the **TM NPU** via trained neural networks
3. Results of the inference are sent from TM NPU to the PLC
4. Function blocks in the PLC evaluate the results

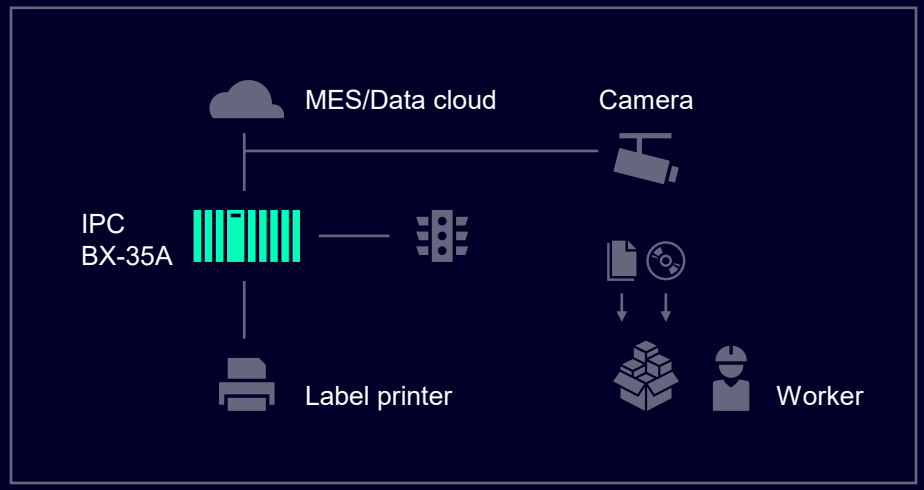


SIEMENS

Use case Vision inspection
Packaging, detection and warning of missing components

Task
- At the end of a PLC production line, the PLC and other components are manually packed into a transport carton
- Main faults are missing parts in the box
- To prevent this from happening, detection and verification is required at several key control points
- Checking the MAC address of each device is necessary manually

Solution
- To solve this problem, an image processing system was implemented in the packaging station that recognizes the MAC address of the device and monitors the packaging process
- The workers receive a red light for each step of the work if parts are missing and a green light if everything is in order. Only then will the label to seal the package be printed



Benefits
- Ensuring Correct Packaging
- No subsequent weighing or similar necessary
- Automatic, error-free recording of device MAC
- Saves time and effort and thus reduces production costs

SIMATIC IPC BX-35A
- Use case of the IPC BX-35A
- With Vision Scenario and Deep Learning Pipeline connection to MES to verify MAC address
- IPC BX-35A
- IFP2200 Basic

SIEMENS

Gracias por su tiempo!

Franco Müller
Técnico Consultor

Siemens S.A.
RC-AR DI FA
Argentina
mailto: franco.muller@siemens.com

www.siemens.com/AI

SIEMENS