AI-GO

No-code Vision AI platform for manufacturing

→ A wide range of pre-trained models
→ High performance even with just a few examples
→ Just a few minutes to go into production
→ Usable even without specific skills

AI-go ADDRESSES COMPLEX INDUSTRIAL PROBLEMS

REQUEST A DEMO!

CLASSIFICATION

APPLICATIONS
Sorting defective or anomalous parts, verifying the presence of components and correct assembly, object detection.

TASK
Predicting the class of an image with confidence score.
AI-go supports two types of models:
- whole image classification: binary (good / scrap) and multi-class (defect type 1 / 2 / n)
- patch-based classification (dividing the image into smaller parts).

ANOMALY DETECTION

APPLICATIONS
Identification and localization of defects or anomalies.

TASK
Precise detection of a defect and generation of a heatmap that allows the machine operator to visualize the “anomalous” area identified by the algorithm, making the neural network’s judgment “explainable”.
AI-go learns to detect defects solely based on images of good parts (unsupervised learning), without the need for examples of scrap parts.

SEGMENTATION

APPLICATIONS
Sorting and differentiation of defects or components and products, analysis of geometries, defect measurement.

TASK
Accurate localization of a defect and generation of a pixel-level mask defining its area, accessible to the line operator for further investigation and validation of the model’s output.
AI-go supports multi-class segmentation models that allow the differentiation of defects based on categories predefined by the operator.

OCR - optical character recognition

APPLICATIONS
Reading labels and codes, especially when a highly accurate performance is required.

TASK
Recognition of printed characters even on irregular surfaces (such as vials, bottles, jars, bags, blisters, and tubes), with non-uniform print quality or embossed and dotted industrial characters.

BENEFITS

PRE-TRAINED MODELS
Automatically select the most effective pre-trained model to solve your problem and specialise it through a few examples.

FAST DEPLOY
Using unsupervised models – which learn from examples of good parts and do not require scrap parts – drastically reduces the time to deploy the vision AI solution from weeks to minutes!

GREAT PERFORMANCE
Maintaining performance over time while ensuring an adequate level of model “explainability”.

USER FRIENDLY
No specific skills are required and it can be used by people without previous knowledge of computer vision, programming or vision AI techniques.

QUICK AND EASY INTEGRATION
Easily integrated with existing vision systems. Are you a vision system integrator? Join the Machine Vision GeeX Program!

VALIDATION AND MONITORING
Manage the validation of new models before deploying them into production, monitor them remotely to ensure they are working properly, and schedule automatic re-training to improve their performance over time.

SMALL-SIZED MODELS
Models are automatically optimised to operate even on devices with reduced computational power (IoT, edge computing).

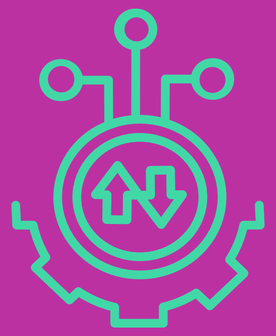
STRONG AND STABLE FOR INDUSTRIAL USE
Designed to operate in industrial scenarios, considering aspects usually ignored by generalist solutions such as cycle time, inference speed and reliability.

ORÒBIX

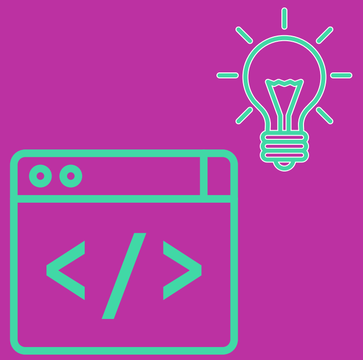
Machine Vision GeeX Program - https://visionaigo.com

5 STEPS to integrate AI-go into your vision application

JOIN THE PROGRAM!

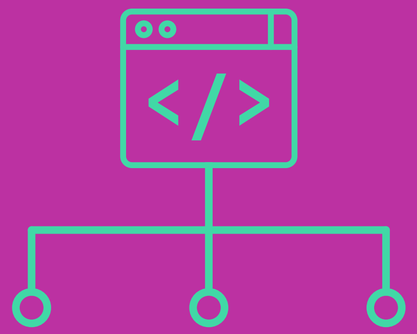


Easy integration



Intuitive functions

```
aigo_attach_runtime
aigo_load_model_request
aigo_image_from_memory
aigo_predict
aigo_get_outcome
```



Same structure for all models

(classification, anomaly detection or segmentation)

5 STEPS to integrate AI-go into your vision application

Intuitive functions

```
// Setup connection AI-go properties
aigo::AigoRuntimeConfiguration runtime_config;
runtime_config.master_url = "169.254.188.190"; //or "localhost";
runtime_config.port = 6379;

// Connect to AI-go runtime!
auto attach_runtime_result = aigo_attach_runtime(context, &runtime_config);
```

1_SETUP CONNECTION

```
// Upload your AI-go model from a given list
aigo_load_model_request(context, "Anomaly_Model_Test_1", runtime_id);
```

2_UPLOAD MODEL

```
// Upload your image!
cv::Mat img = cv::imread("my_image.bmp");
aigo::AigoImage image = aigo_image_from_memory(img.data() 3 * img.rows() * img.cols()));
```

3_UPLOAD IMAGE

```
// Build input
aigo::AigoInputData input;
input.image = img;
input.model = "Anomaly_Model_Test_1";

//Performe an inference!
auto id_result = aigo_predict(context, input.get(), 1, runtime_id);
```

4_INFERENCE

```
//Get outcome
auto output_result = aigo_get_outcome(context, id_result.execution_id, 5000, runtime_id);

//////////////////////////
// Serialize output image
auto segmap = output_result.output_datas[0].segmentation;
```

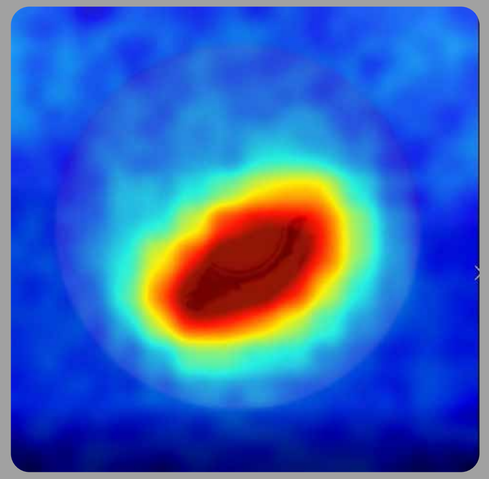
5_GET OUTCOME

Same structure for all models
(classification, anomaly detection or segmentation)

```
// ANOMALY
heatmap = aigo_get_heatmap(segmap);

// Decode the heatmap
cv::Mat imgHeatmapGray = cv::Mat(_segmap->height, _segmap->width, CV_64FC1, heatmap.pointer);

// Convert to 1 channel image
imgHeatmapGray.convertTo(imgHeatmapGray, CV_8UC1, 255.0, 0.0);

// Blend with original image
cv::applyColorMap(imgHeatmapGray, imgResult, cv::COLORMAP_JET);
double alpha = 0.85;
cv::addWeighted(imgResult, alpha, imgOriginal, (1.0 - alpha), 0.0, imgResult);
```

OpenCV

INPUT



OUTPUT

ANOMALY DETECTION

1
Get the result
with OpenCV
(or similar library)

2
Blend the result map with the original image, producing a **heatmap**

Same structure for all models
(classification, anomaly detection or segmentation)

```cpp
// SEGMENTATION
// Decode the binarymap
auto binarymap = aigo_get_binarymap(segmap);
imgBinMap = cv::Mat(segmap->height, segmap->width, CV_8UC1, binarymap.pointer);

// Make binarymap RGB and blend with original _image
std::vector<cv::Mat> channels(3);
channels[0] = cv::Mat::zeros(imgBinMap.rows, imgBinMap.cols, CV_8UC1);
channels[1] = imgBinMap;
channels[2] = cv::Mat::zeros(imgBinMap.rows, imgBinMap.cols, CV_8UC1);
cv::merge(channels, imgResult);

double alpha = 0.5;
cv::addWeighted(imgResult, alpha, imgOriginal, (1.0 - alpha), 0.0, imgResult);
```

INPUT

OUTPUT

SEGMENTATION

1
Get the result
with OpenCV
(or similar library)

2
Overlay the map on the original image, **highlighting the segmented area**

ORÒBIX

Machine Vision
GeeX Program

CONTACT US!

Orobix srl
via Gabriele Camozzi 144
24121 Bergamo - Italy
+ 39 035 017 0561

www.orobix.com
info@orobix.com