Quality Inspection Cell:

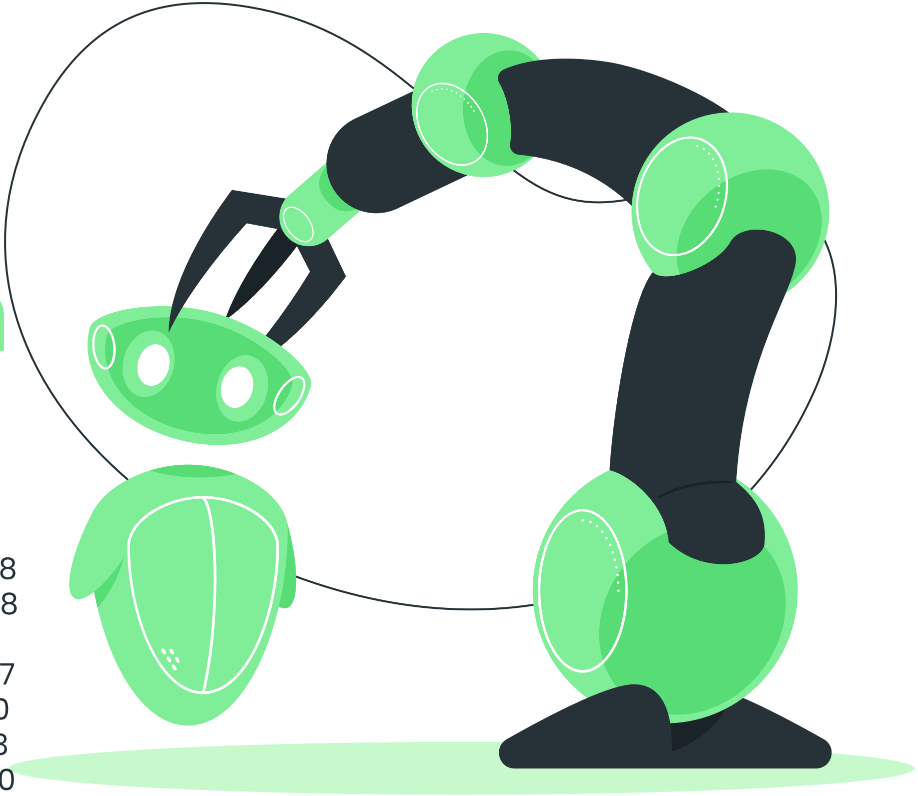
Burrs detection

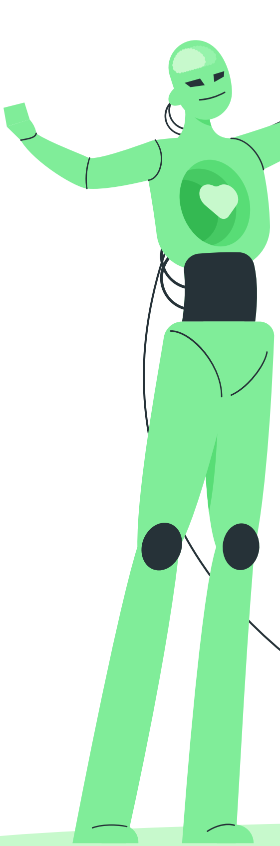
Mechatronic Design MR3009

P4.

José Angel Soto Hernández	A01039978
Nathalie Vilchis Lagunes	A01364838
Hector Everardo Martínez Cisneros	A01282300
Teclo Moreno Rodriguez	A01252067
Estefany Morales Valdes	A01281880
Diego A. Santisteban Pozas	A01154423
Jose Antonio Arrambide Garza	A00817790

24/11/2021





1
Original Problem/opportunity definition

Opportunity definition

Original Problem

During the drilling in aluminum process the presence of burrs is likely to happen.

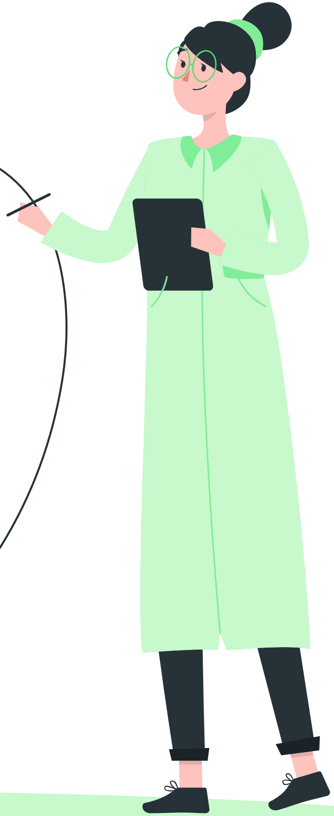
Burrs might represent a high risk in the subsequent process of the product

Suggested solution

Automate the inspection of drilled aluminum profiles. Integrate machine learning, computer vision and cobot usage to sort profiles (accepted and rejected parts). Eliminate human interaction.

2

Value proposition conclusion



Original Value Proposition

Quality Inspection Cell: Burrs detection

Increase

- Adaptability
- Detection speed
- Accuracy
- Higher quality

Create

- User-friendly interface
- Database
- Alarms
- Machine learning

Reduce

- Price
- Workspace
- Hardware/Software requirements

Eliminate

- Overhead
- Human error

Value Proposition Conclusion

Achieved

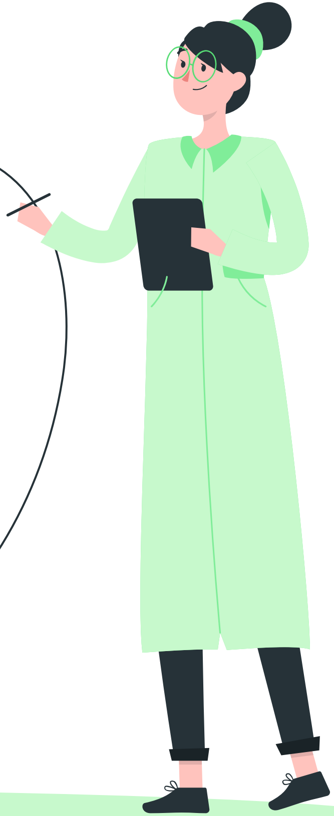
- High detection accuracy
- Higher quality products,
- High adaptability
- Machine Learning
- Result Database

NOT Achieved

- Eliminate all human interaction
- Real-time results
- User friendly interface
- Reduction of workspace
- Reduction of hardware and software requirements

3

Final Product Requirements



Final Product Requirements

Achieved

- Non invasive operation
- Cobot-Python-Orange communication
- Burr classification (92.2 %)
- Creating knowledge (databases)
- Machine learning model (Orange)

Partially Achieved

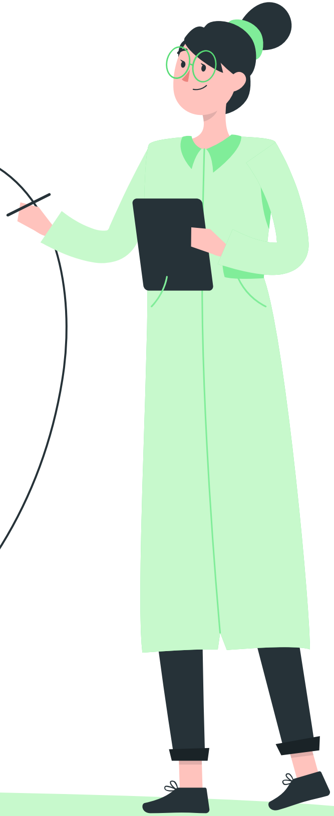
- Reduce workstation accidents
- Working station integrated in single working table (Simulated)
- Reduce workforce (simulated - 1 person)

NOT Achieved

- Near zero human interaction
- Response time less than 5 seconds
- Avoid product damage
- Avoid infrastructure damage
- Real time CV

4

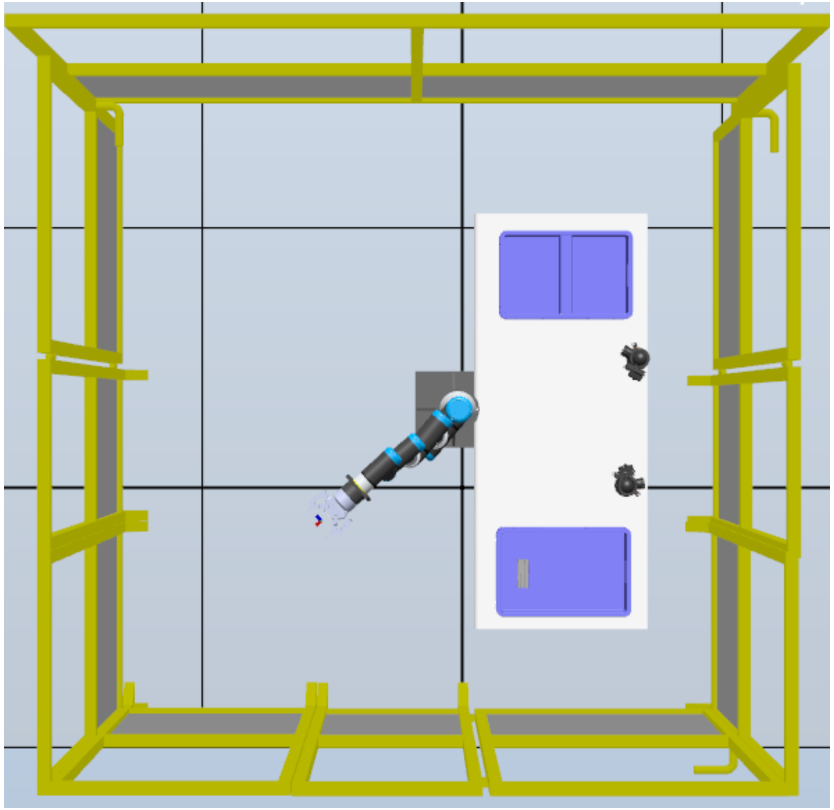
Final Design Details

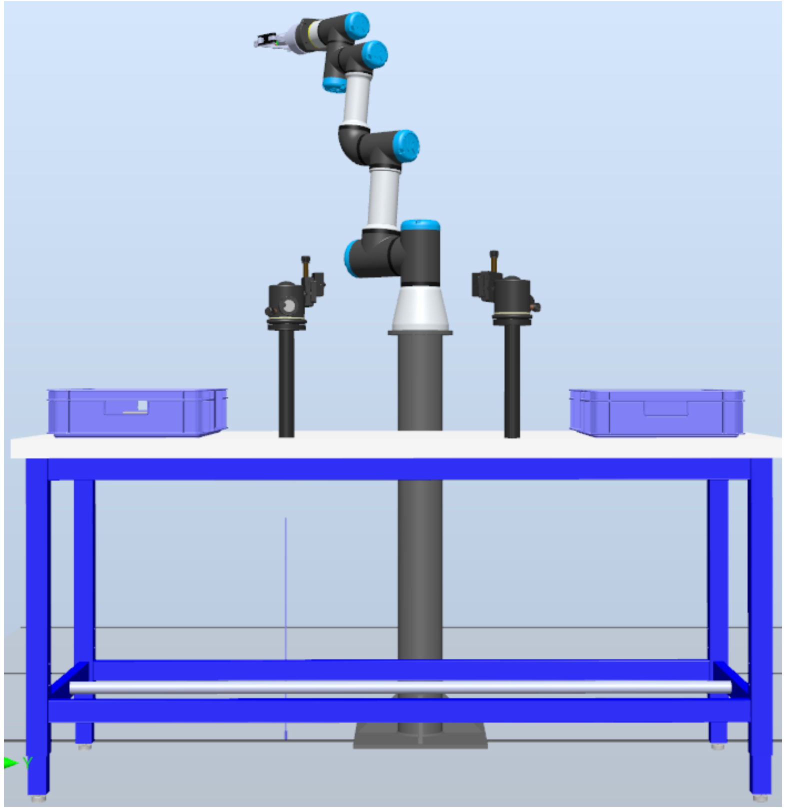


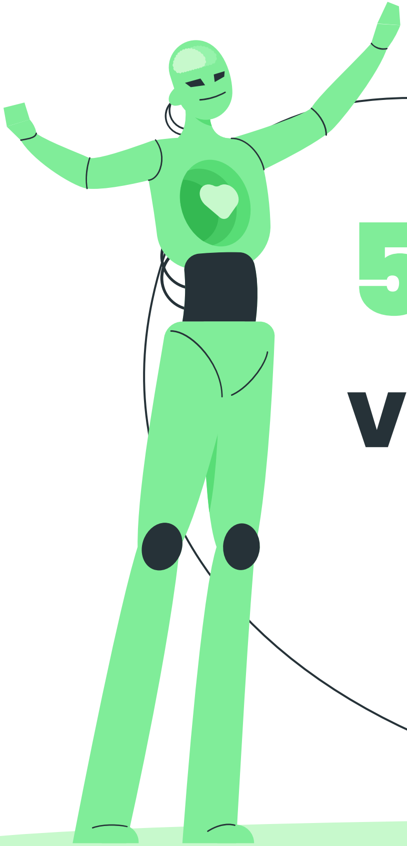
Final Design details

- Final model is fully simulated in RobotStudio
    - UR3 CAD model imported to RobotStudio
    - Custom controller designed for six axis robot with an ABB 140 driver and Robot Flex 125 toploader mechanical template.
- Communication done through python
    - Robotstudio -> Python -> Orange -> Python -> Robotstudio
- Classification and predictions done in orange

Cobot station

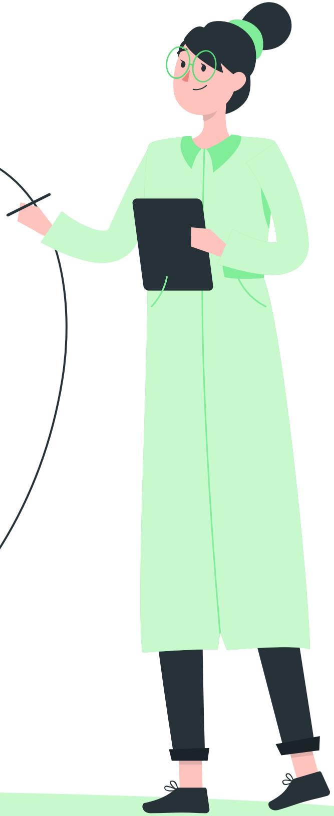






5

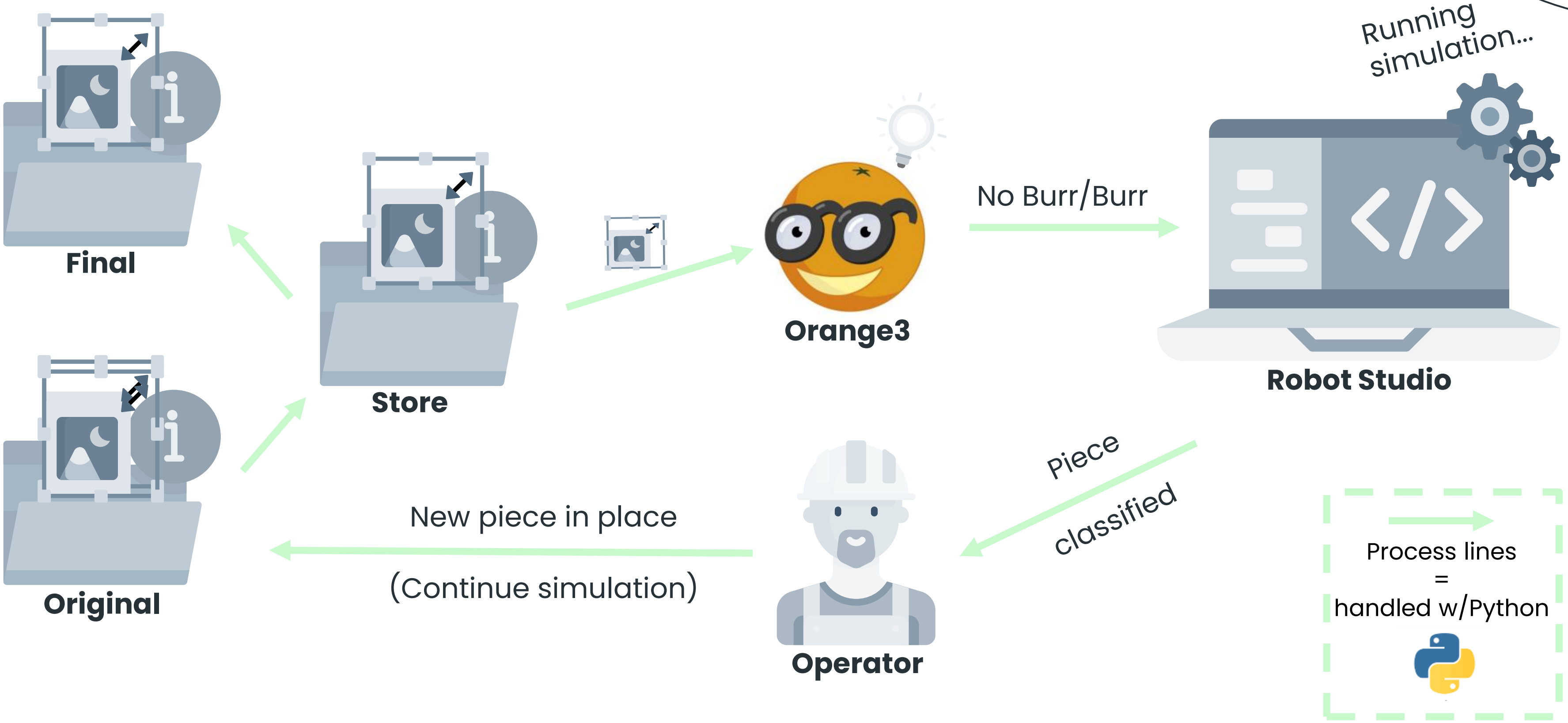
Verification results



Verification results

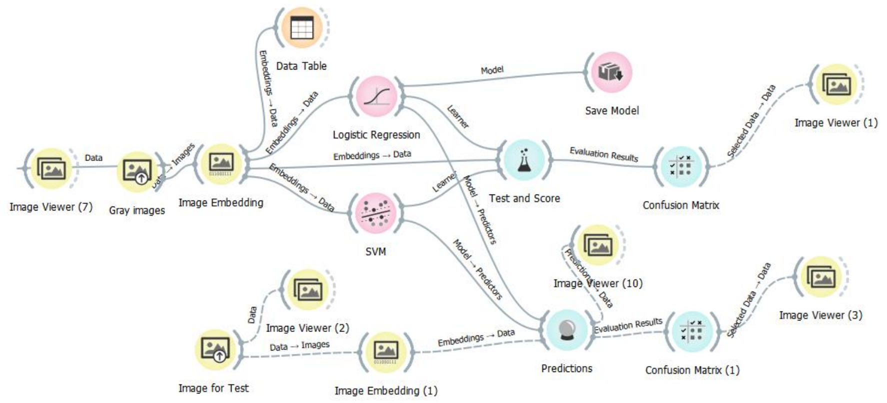
1. **Testing plan**
    1. Classification
        - We ran several tests with assorted images with burrs and no burrs
        - Each image went from Original folder -> Stored folder -> Final folder
        - Orange reads image from stored folder
        - Prediction from image is sent from python to Robotstudio, cleans stored folder places the used image in final folder (to prevent repetition)
        - UR3 takes a pre programmed path according to the classification results
        - UR3 resets to home position and python communication is closed
        - Test concludes when the simulation places the profile in the correct bin and the program resets.

Prototype Details: Process Loop

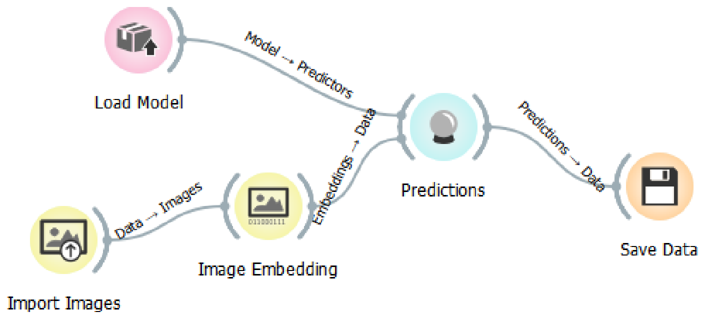


Simplified Orange program

Original



Simplified



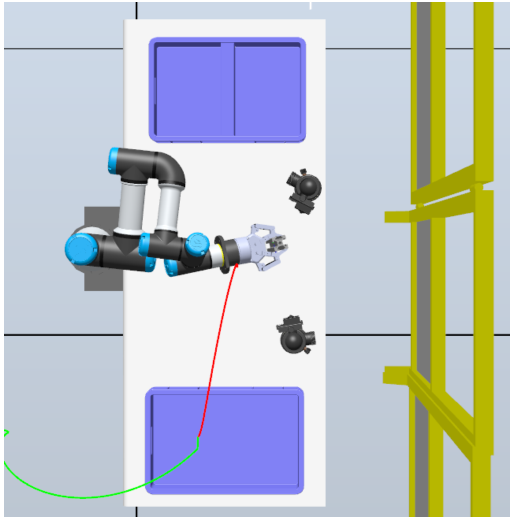
- Logical regression training model was selected (Better results than SVM and random Forest) AC > 92%
- Grayscale database was used for training

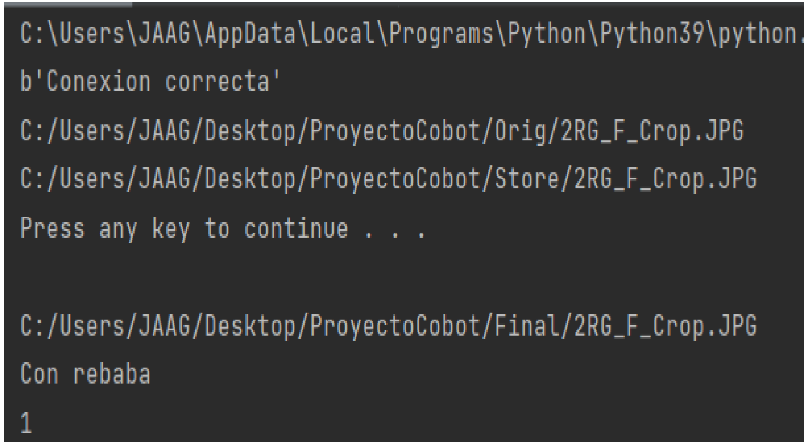
- Logic regression model is loaded into new orange file
- Picture saved by python in Store folder is loaded for classification
- Much faster orange results
- No need to retrain every new run

Python – Robotstudio connection and initiation of python script

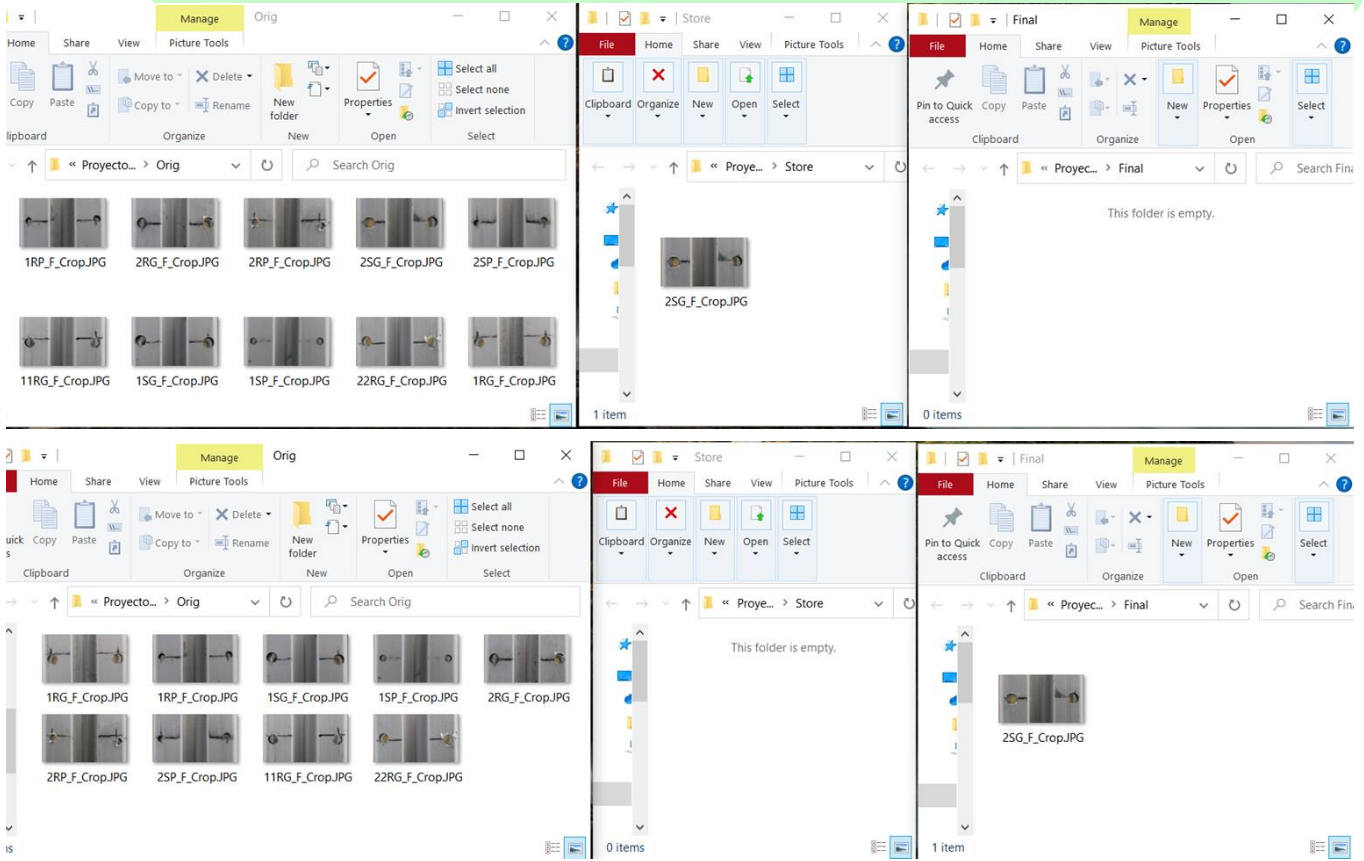
1. Simulation starts in RobotStudio
2. RobotStudio creates server host; waits for python connection confirmation.
3. Cobot takes path according to classification results

1. Once confirmed, python runs image filter script
2. After Orange finishes image analysis, user needs to press any key to continue
3. Classification results will be stored in a database
4. Python reads and sends results to Robotstudio





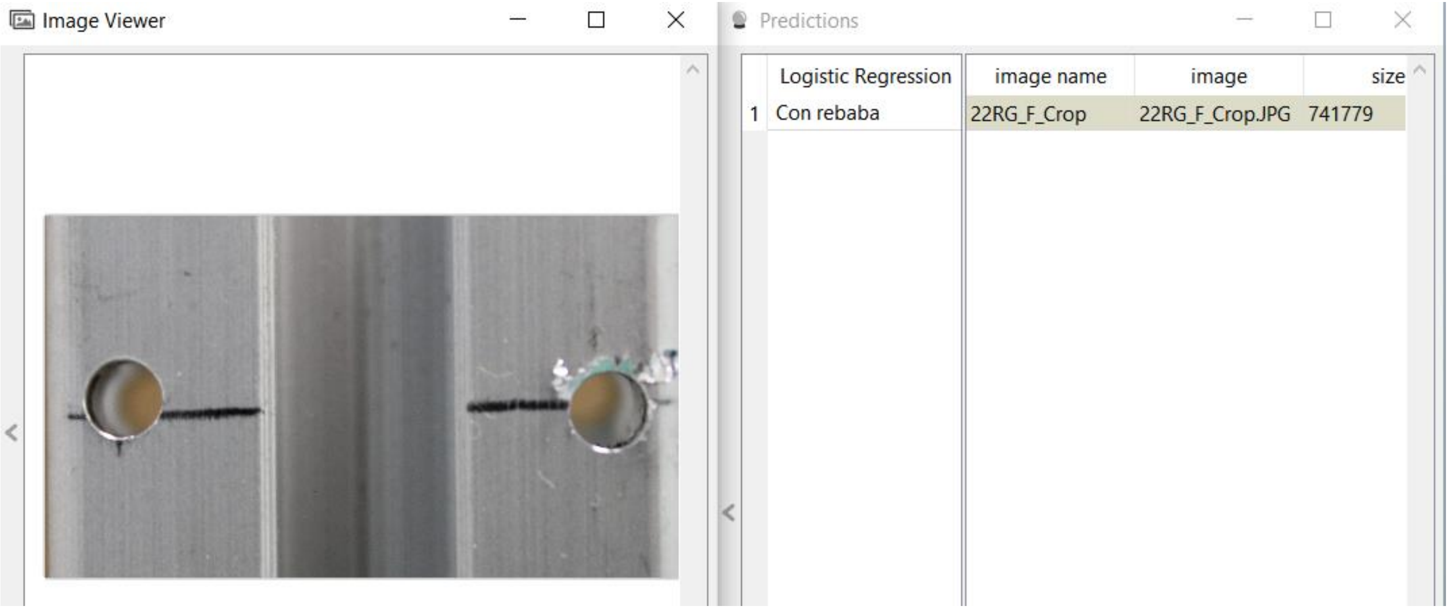
Python image manipulation



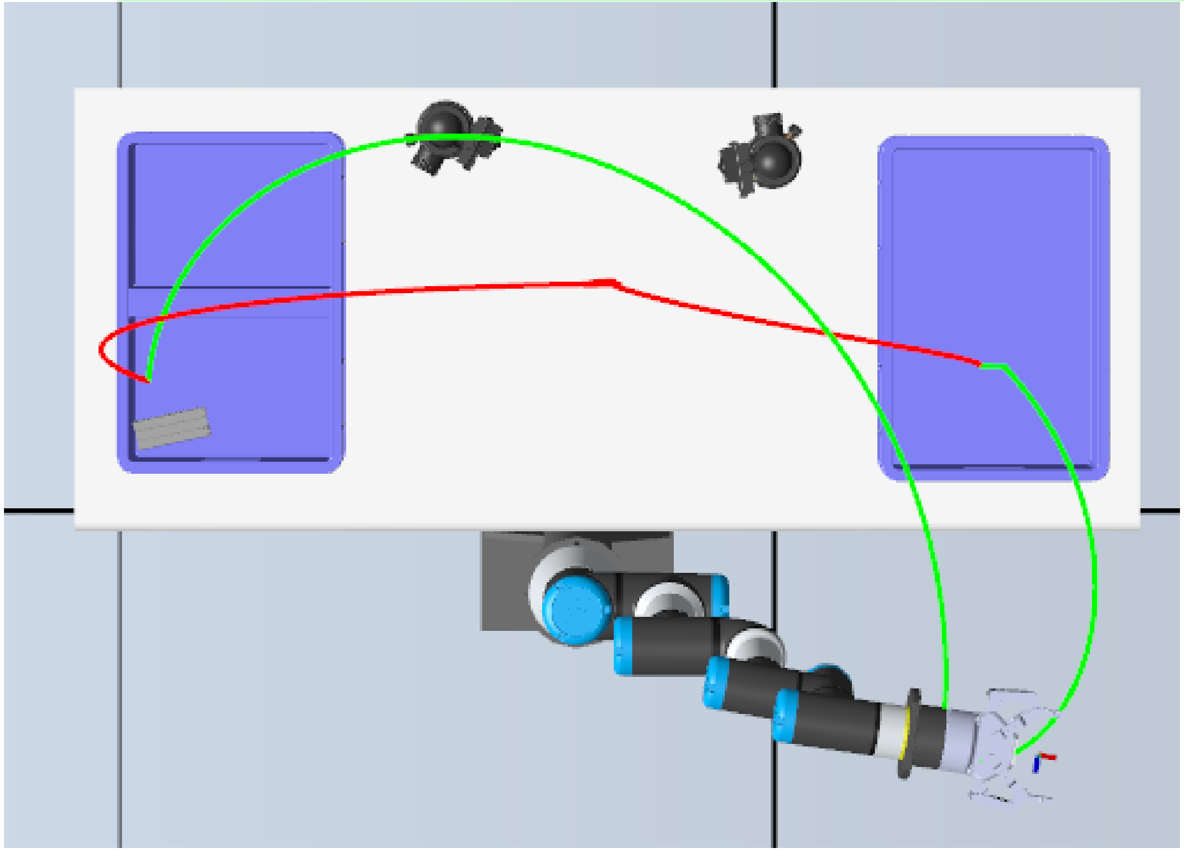
1. Orig folder holds pictures to analyze
2. Python extracts a random picture from Orig and moves it to Store folder
3. Orange reads picture from Store folder
4. Python moves picture from store folder to Final folder *This is to prevent image repetition

Orange single image results

- ❑ As it can be seen Logistic Regression model gives a correct prediction
- ❑ The prediction result is stored in a database and read by python
- ❑ RobotStudio receives prediction result from python



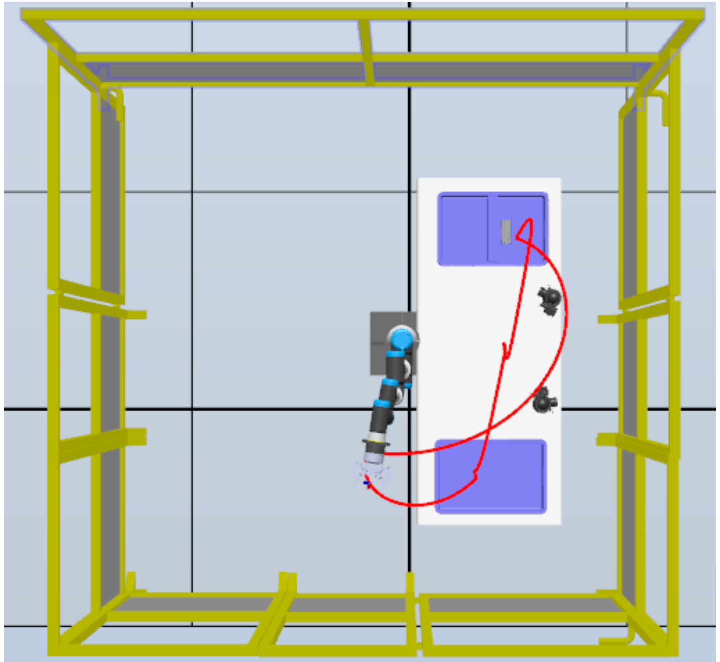
Cobot path for image with burrs



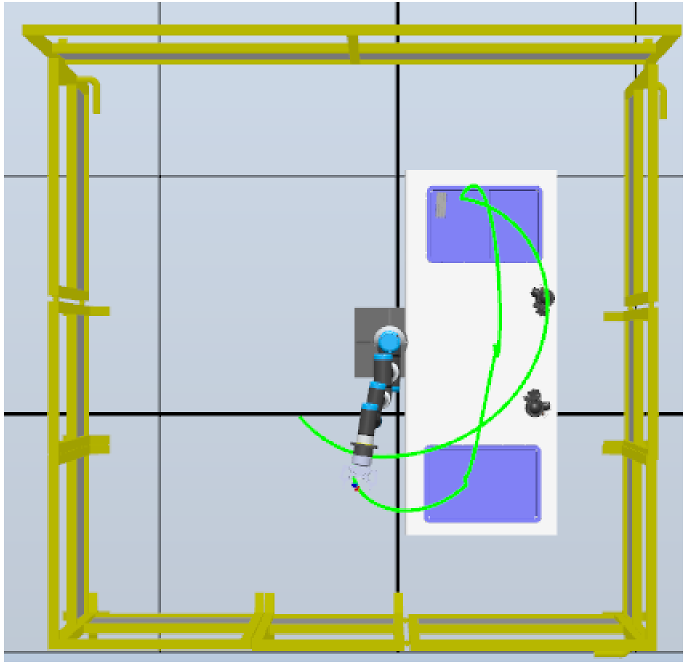
- Cobot wont move from classification area (between cameras) until it receives Orange's result from python
- Red path means the cobot is holding the profile
- After leaving profile the Cobot returns to its original location.

Pre programed paths

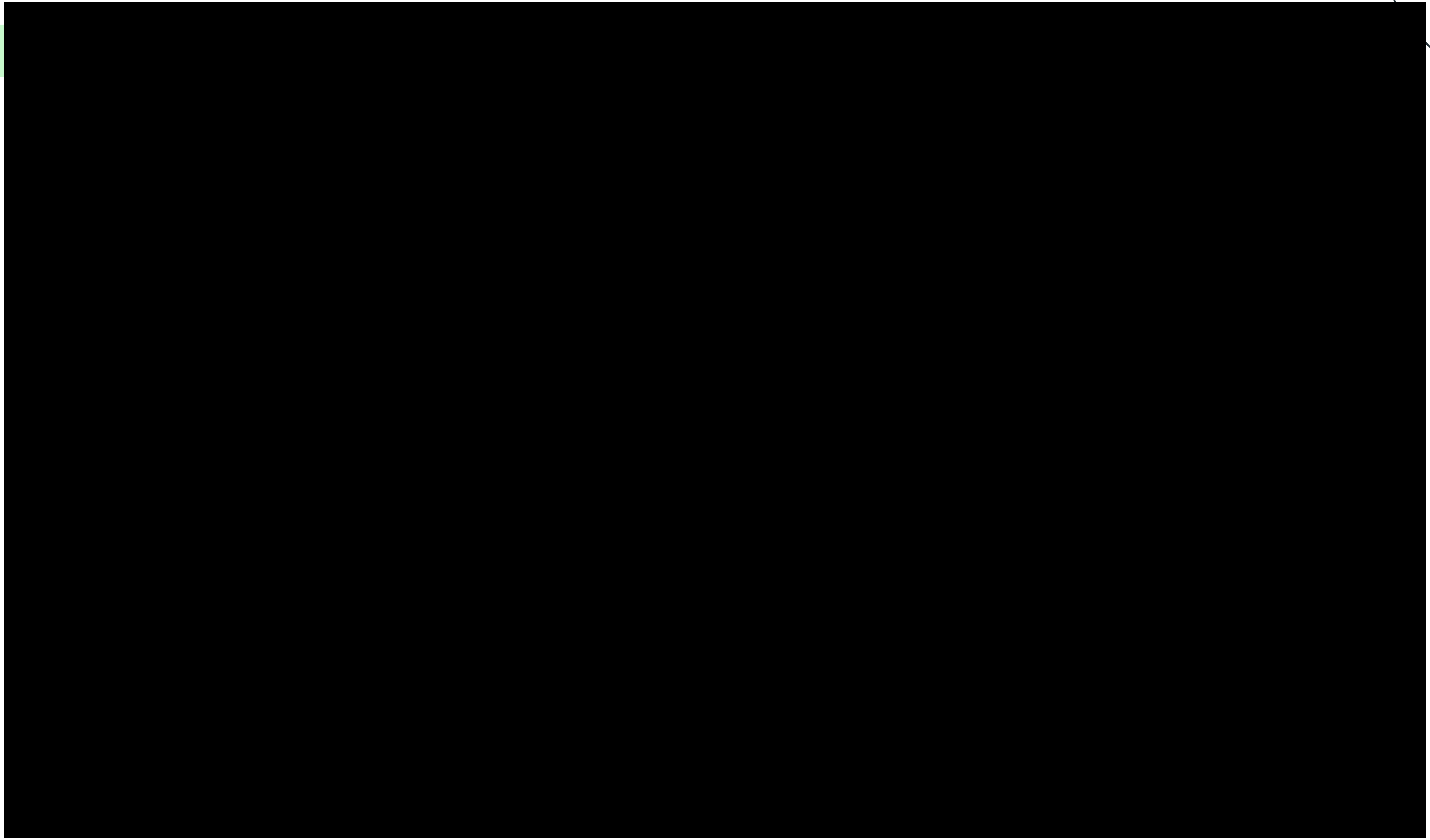
Path with no burrs



Path with burrs



Video test results



Test results

- The model of classification of images with or without burrs in orange is of **92.9%**

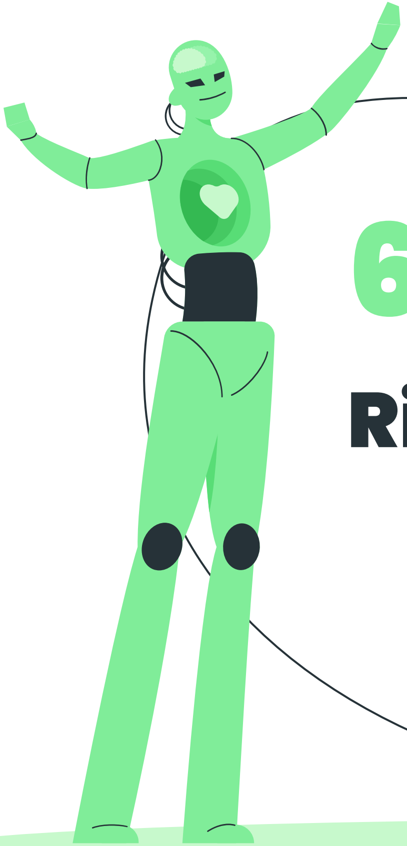
- Times of simulation experiment results:

Steps/Velocity	Slow	Medium	Maximum
Pick up and position the piece	14s	11s	9s
Python y Orange	35s	23s	23s
Position the piece in the bins	14s	10s	8s
Total Time	63 s	44 s	40 s

Orange test results

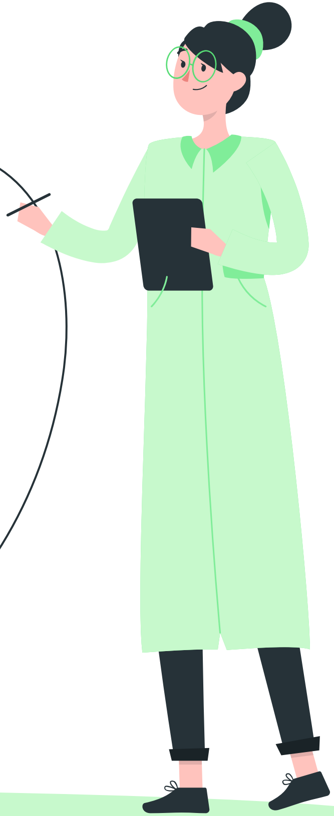
image name	image	Logistic Regression	Logistic Regression (Con rebaba)
string	string	Con\ rebaba Sin\ rebaba	continuous
meta	meta origin=C:/Users/JAAG/Desktop/ProyectoCobot/Store type=image	meta	meta
11RG_F_Crop	11RG_F_Crop.JPG	Sin rebaba	0.318308626
1RG_F_Crop	1RG_F_Crop.JPG	Con rebaba	0.879055988
1RP_F_Crop	1RP_F_Crop.JPG	Con rebaba	0.529156108
1SG_F_Crop	1SG_F_Crop.JPG	Sin rebaba	0.001326119
1SP_F_Crop	1SP_F_Crop.JPG	Sin rebaba	2.02E-08
22RG_F_Crop	22RG_F_Crop.JPG	Con rebaba	0.99999972
2RG_F_Crop	2RG_F_Crop.JPG	Con rebaba	0.999999126
2RP_F_Crop	2RP_F_Crop.JPG	Con rebaba	0.999826744
2SG_F_Crop	2SG_F_Crop.JPG	Sin rebaba	0.024250907
2SP_F_Crop	2SP_F_Crop.JPG	Sin rebaba	0.016766824

*Testing carried out with the original images flipped 180°



6

Risk Assessment



Risk Assessment

Major Risks	Reliability/ Performance	Fast Process
	R1: Wrong classification of the aluminium profiles	FP1:Takes more than 5 seconds to classify
	R2: Software failure	FP2: Process stops due to bad connectivity
	R3: Requirement of multiple software programs management	FP3: Maintenance and upgrades take too much time
	Costs	**Safety**
	C1: Operational cost are bigger than revenues	S1: Possible accident that injures the user
	C2: Initial investment is very high	S2: Possible accident that damages the Cobot
	C3: Maintenance of system expenses	S3: Possible accident that damages the aluminium profile

Risk Assessment

			Business Impact				
			Extreme	Major	Moderate	Minor	Insignificant
			100%	80%	62%	25%	1%
Occurrency	(Almost) Certain	100 %		FP1			
	Probable	80%					
	Possible	62%	C1	FP2			
	Unlikely	25%	S1	R2, C2,S2	R1	S3	
	Rare	1%		R3,FP3	C3		

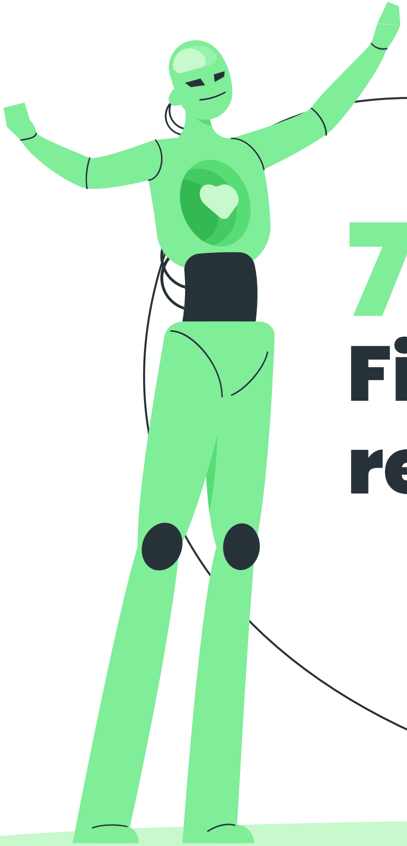
Tolerate	to Improve	Must be corrected

Mitigation Plan

	Reliability/ Performance	Risk	Actions to eliminate/mitigate Risk
Major Risks	R1: Wrong classification of the aluminium profiles	M	Improve the quality of the image acquisition.
	R2: Software failure	M	Regular check-ups to verify that the softwares are in optimals conditions
	R3: Requirement of multiple software programs management	L	Run the programs in a device that supports all the softwares running at the same time without saturating the RAM.
	Costs		
	C1: Operational cost are bigger than revenues	M	Try to cut unnecessary costs and match it with the sales prices.
	C2: Initial investment is very high	M	Try to cut unnecessary costs to low initial investment.
	C3: Maintenance of system expenses	L	Make a study of how often are maintenance shall be done.

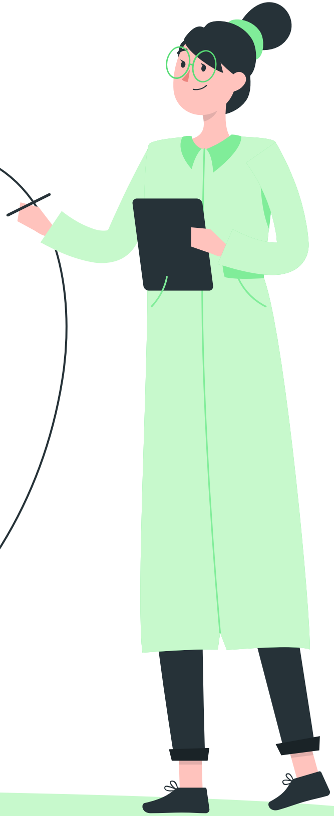
Mitigation Plan

	Fast Process	Risk	Actions to eliminate/mitigate Risk
Major Risks	FP1:Takes more than 5 seconds to classify	H	Optimize the movement path and velocity of the robot.
	FP2: Process stops due to bad connectivity	M	Implements communication protocols to avoid interference.
	FP3: Maintenance and Upgrades take too much time	L	Program maintenance when production line is not running. (at night or holidays)
	Safety		
	S1: Possible accident that injures the user	M	Redesign the working area to avoid accidents, Integrate a Stop Button
	S2: Possible accident that damages the Cobot	M	Redesign the working area to avoid accidents, Integrate a Stop Button
	S3: Possible accident that damages the aluminium profile	M	Ensure by testing that the robot is able to place the aluminium profile in a correct place



7

Final IP recommendations



Claims and IP Instruments

Copyright

Orange algorithm for the detection of burrs in aluminum profiles.

This patent protects the algorithm created to analyse compare and classify provided images for the detection of aluminum burrs. It also includes a machine learning stage to improve accuracy and precision through a previously loaded database.

Copyright

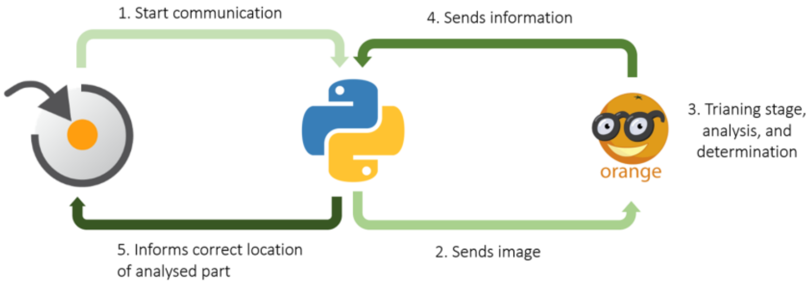
Python algorithm used as an intermediary for the synchronization between orange and robot studio.

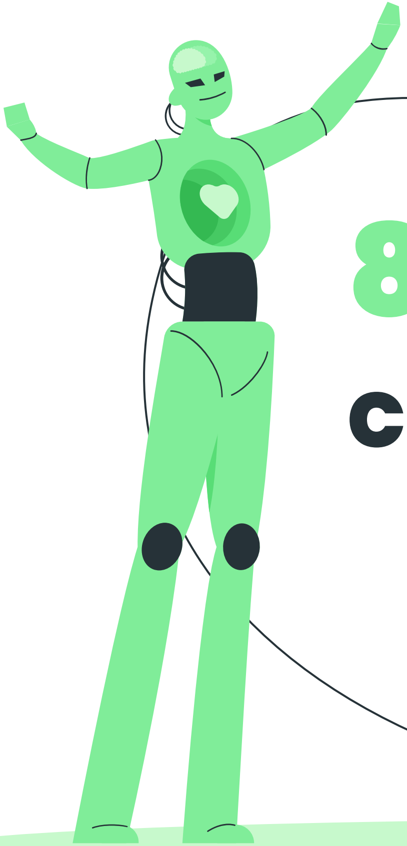
Algorithm that manipulates a robotstudio simulation with information previously processed by orange. It provides information to orange to complete the image classification process, which is sent back to python. This information is used to determine the specific placement of the aluminum profiles in robotstudio.

Patent

Computer vision device and quality inspection cell for burr detection in aluminum profiles.

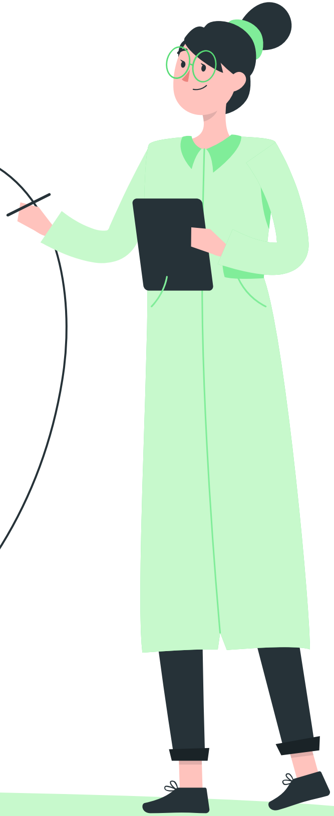
This is an inspection cell that incorporates a cobot and computer visión to fully automate the classification of metal burrs. This inspection cell takes into account the pick of the part, the inspection of the profile in an specific area and the correct distribution of the aluminum part.





8

Conclusions



What's next

Make a physical prototype based on the simulation. (Implementation of the product).

Validation tests (On field testing for communication between softwares, robot trajectory, image capturing, etc.).

Optimization of classification model to reach more than or equal to 95% detection accuracy.