POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

EXECUTIVE SUMMARY OF THE THESIS

Artificial Intelligence for Image Classification and Anomaly Detection in the Food Sorting industry: a Comparative Study

TESI MAGISTRALE IN MECHANICAL ENGINEERING – INGEGNERIA MECCANICA

AUTHOR: Pietro Oppici, Jacob Niccolai

ADVISOR: Prof. Marco Tarabini

CO-ADVISORS: Prof. Marco Bocciolone, Ing. Davide Maria Fabris

ACADEMIC YEAR: 2021-2022

1. Introduction

This thesis aims at the implementation and comparison of machine vision algorithms based on deep learning for the quality inspection of vegetables. It is part of the project "*Studio e sviluppo di TEcnologie avanzate per il SORting automaticO nei processi di produzione alimentari – TESORO*", issued by the Italian Ministry of Economic Development and in collaboration with various industrial partners. Among the project partners there is Raytec Vision S.P.A., which is a leading expert in fruit and vegetables optical sorting. The work is motivated by the fact that it is not possible to classify tomatoes with the same traditional techniques used for other food types and by the interest in the investigation and development of artificial intelligence solutions. The scope of this work is to understand whether deep learning algorithms based on neural networks [1] would be a suitable solution to perform the task, possibly reaching market-ready performances.

In particular, both Image Classification (IC) and Anomaly Detection (AD)-based models are studied to find the best solution for quality control on tomatoes. In this manner an automated system able to substitute human labor in that procedure is made possible. Three state-of-the-art AD algorithms and three IC algorithms based on neural networks are built and compared in a statistically significant manner. Finally, the most suitable one is selected. Additional experiments are performed to test the selected model performances with in-field acquired data. In addition, strengths and weaknesses of the other models are highlighted in the process because they could be exploited in similar applications with different priorities.

In conclusion, a robust solution to the industrial problem is achieved. The developed software was integrated on already existing sorting hardware without any major modification.

The document is structured as follows. The literature research section shows the outcome of the study on related work. The suitable models are identified here. The methods section describes the work needed to adapt those models to the specific case, the metrics chosen to evaluate the performances, the model optimization and the tools used to compare their performances. The results section is the direct consequence of the methods applied on the given data. The conclusions section will highlight the main

findings of the work, proposing further future developments.

2. Literature research

The first step in the development of this thesis was the literature research, aimed at the identification of the most suitable algorithms for the specific application. In particular, the procedure for the selection of the IC algorithms was the following. Firstly, a cross-reference with the Raytec Vision S.P.A. engineers about what algorithms were implemented in their commonly used software (i.e., MVTec Halcon®) was performed. Then, the equivalent of these algorithms in the form of Python open-source software was searched. In addition to that, research on academic and commercial databases was performed using the following keywords: python, deep learning, IC, anomaly detection. Relying on the afore-mentioned rules, three neural-networks based algorithms were identified for IC [2]:

- Mobilenet
- Resnet50
- GoogLeNet

As for the choice of the deep AD algorithms, three state-of-the-art models were identified in literature. These are:

- CFLOW-AD
- PatchCore
- DFKDE

Raytec Vision S.P.A. has not yet approached AD at all but is interested in performing a comparison among AD and IC algorithms.

3. Dataset

Two datasets were provided by Raytec Vision S.P.A. The first is composed of 3370 images with the following distribution:

- 974 samples of good, peeled tomatoes
- 1000 samples of bad, yellow tomatoes
- 1000 samples of bad, green tomatoes
- 396 samples of not correctly peeled tomatoes and affected by anthracnose

These were grouped in:

- 974 good samples
- 2396 bad samples

These images are acquired in controlled environment and in proper lighting conditions, those are the standard the machine works with, in possible future market applications. Some examples extrapolated from this dataset can be seen in Figure 1.

The second dataset consists of 252 images of peeled tomatoes and 228 images of unpeeled tomatoes, labelled as good and bad. These images were acquired with a smartphone in uncontrolled lighting conditions. Moreover, in some pictures, even unwanted objects are present. It must be noticed that the number of images here was quite low. For these reasons this dataset was used in the following to test the robustness of the proposed system.

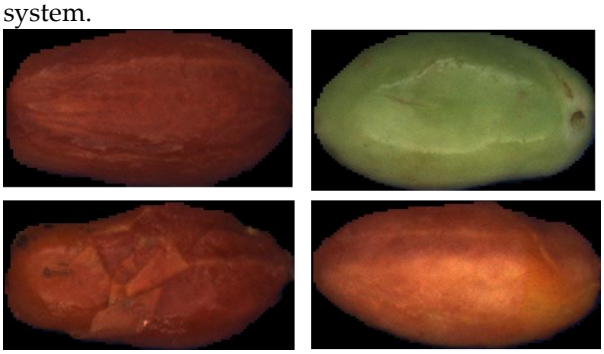

Figure 1 - Examples from the first dataset: on the top left a correctly peeled tomato can be seen, top right a green one, bottom left a badly peeled one and on the bottom right a yellow one

4. Method

In this section the workflow followed to reach the final software configurations and the experimental procedures used to optimize and compare the models are reported. A black-box scheme was implemented to handle the models. The inputs are the images and the output is the sorting classification result. The different models are the content of the black box. To do so YAML file containing all the configurable parameters was used. Changing the parameters contained in these files, it was possible to train and compare the different models without the need of hard coding every time the algorithm changes. This standardized approach was helpful to compare the models and speeds up the proceedings.

To reach this structure for the IC-based models, Tensorflow and Keras were two fundamental libraries used which permitted to exploit transfer learning[3]. The neural networks considered were taken in their pre-trained form on Image-net dataset and then modified for the given task. For the AD-based algorithms the fundamental library used was Anomalib.

To evaluate the models the following metrics were chosen:

- Inference time
- Area Under the Receiver Operator Curve (AUROC)
- Accuracy

The inference time was chosen as a metric since the decision-making speed plays a huge role for the application this thesis focuses on. Accuracy and AUROC were chosen to evaluate the quality of the sorting procedure in terms of ability to discern good samples from bad ones. A Hyper-Parameters Optimization (HPO) procedure was then conducted. In this framework, the black box is the code in charge of optimizing the hyper-parameters. It receives as input a configuration file in YAML form containing the hyperparameters to tune, along with their range of values, and it gives as output the optimized value for each parameter. The optimization results can be visualized in the form of parallel coordinates plots, like the one reported in Figure 2.

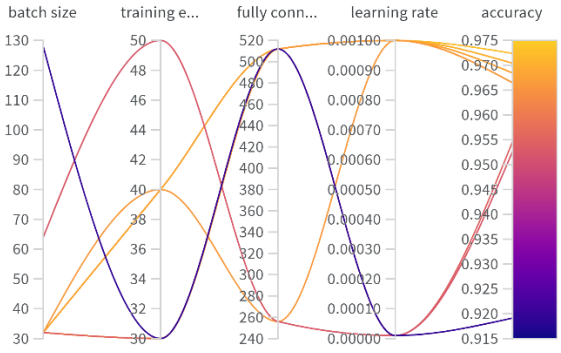

Figure 2 - Example of parallel coordinates visualization of the Hyper Parameters Optimization

After ensuring every model was optimized for the given task, the comparison among them could be made to find the most suitable one for the industrial application. To generalize the results, the K-Fold cross-validation method was applied. The K performances values obtained were treated as a set of scores and a significance test was set up among them. To determine whether the collected set of scores was large enough to allow for reliable significance testing, bootstrap power analysis was used, as proposed by Yuan et al. [4]. On this set of K scores for each model and each metric Almost Stochastic Order (ASO) was applied, as proposed by Dror et al [5], to find if there were significant evidence of dominance of a certain model over another. By the methods listed above, the models were compared to find the best performing one.

5. Results

Having performed HPO on each model it was possible to evaluate them with the typical train-validation-test procedure to have a first guess of the performances. To confirm the correct functioning of the model in this phase, the Gradient-weighted Class Activation Mapping (Grad-CAM), proposed by Selvaraju et al. [6], was used where possible. An example is shown in Figure 3. Grad-CAM is a technique used to visualize which regions of the input are "relevant" to perform the prediction.

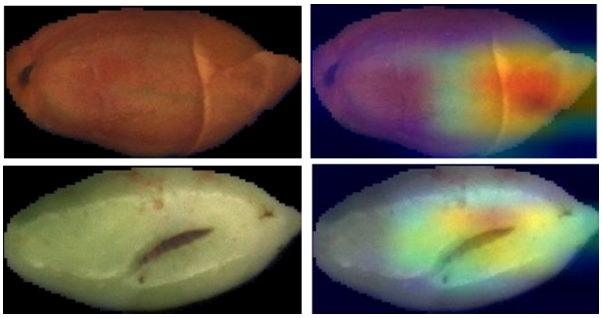

Figure 3 - Examples of Grad-CAM: the model is focusing on the correct portions of the image in which the defect is. This indicates that it is working correctly

The K-Fold was then applied. Via the power analysis it was shown that K=5 was sufficient to perform reliable significance testing on all models. Results of the cross-validation procedure are reported in Table 1, Table 2 and Table 3.

Table 1 - Accuracy results of the different models

Accuracy [%]	Fold n°1	Fold n°2	Fold n°3	Fold n°4	Fold n°5	Mean	Std. deviation
ResNet50	99.55	98.81	99.41	98.37	99.55	**99.14**	$4.7*10^{-1}$
Mobile Net	98.22	98.51	98.66	98.52	99.11	98.61	$2.9*10^{-1}$
GoogLeNet	98.48	96.59	97.92	97.33	97.92	97.64	$7.2*10^{-1}$
CFLOW-AD	95.34	94.94	95.62	95.22	95.45	95.31	$\mathbf{2.6*10^{-1}}$
PatchCore	92.12	95.27	95.11	93.53	96.32	94.47	1.65
DFKDE	84.09	82.51	82.51	83.91	83.39	83.28	$7.5*10^{-1}$

Table 2 - AUROC results of the different models

AUROC [%]	Fold n°1	Fold n°2	Fold n°3	Fold n°4	Fold n°5	Mean	Std. deviation
ResNet50	100	100	100	100	99.99	**99.99**	$\mathbf{1.69*10^{-3}}$
Mobile Net	99.99	99.98	99.99	99.99	99.73	99.94	$1.15*10^{-1}$
GoogLeNet	100	100	100	100	99.64	99.93	$1.42*10^{-3}$
CFLOW-AD	95.70	94.22	95.93	94.97	94.81	95.13	$6.93*10^{-1}$
PatchCore	97.40	98.43	98.77	96.62	99.01	98.04	1.02
DFKDE	93.09	90.25	89.97	91.41	91.08	91.16	1.23

Table 3 - Inference time per image results of the different models

Inference time per image [ms]	Fold n°1	Fold n°2	Fold n°3	Fold n°4	Fold n°5	Mean	Std. deviation
ResNet50	3.5	3.7	3.6	3.6	3.8	3.6	$5.7*10^{-1}$
Mobile Net	1.7	1.3	1.3	1.3	1.4	**1.4**	**$1.7*10^{-1}$**
GoogLeNet	2.6	2.2	2.2	2.3	2.3	2.2	$2.5*10^{-1}$
CFLOW-AD	28.5	30.3	25.6	32.2	27.7	28.9	2.3
PatchCore	17.5	15.8	15.6	16.6	14.6	16.0	1.1
DFKDE	11.7	13.1	14.9	13.5	14.0	13.4	1.2

The inference times refer to results obtained with a NVIDIA A100 GPU. After that, by means of ASO the models are compared with a confidence level of 95%. These comparisons bring the following results. In terms of sorting quality, it is possible to list the models from the best to the worst as follows:

1. ResNet50
1. MobileNet
2. GoogLeNet
3. PatchCore
4. CFLOW-AD
5. DFKDE

In terms of inference speed, it is possible to list the models from the best to the worst as follows:

1. MobileNet
2. ResNet50
3. GoogLeNet
4. DFKDE
5. PatchCore
6. CFLOW-AD

Given the need to integrate the here presented algorithms on existing machines, the industrial partner suggested that the almost doubled sorting speed achieved by MobileNet compared to ResNet50 represents a big advantage for which it is acceptable to sacrifice a few percentage points in accuracy and AUROC. Following this reasoning MobileNet was chosen as the best model.

5.1. Additional Results

This paragraph is dedicated to some insights. In particular, two main topics are treated:

1. Binary IC on the second dataset to study the robustness of the system
2. Multi-class IC

Binary classification on additional dataset

The aim of the following experiment is to verify the robustness of the system. To this end it was studied if the MobileNet-based model would have been able to operate in a satisfactory manner also when using the second dataset, containing in-field acquired images with elements of disturbance. In addition, this dataset has relatively few samples which is generally a problem in deep learning settings. Still, the proposed solution was able to perform the sorting procedure with the performances reported in Table 4.

Table 4 - performances on MobileNet on the in-site acquired low quality dataset : this shows how robust the system is

	Accuracy [%]	Inference time per image [ms]	AUROC [%]
Values	98.67	1.9	98.57

To further test the robustness of the proposed solution, the same IC procedure was repeated with random background images like the ones in Figure 4. This change did not show any significant influence on the results.

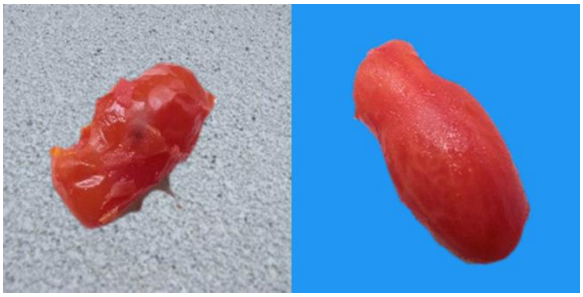

Figure 4 - Example of images with random background used to test the robustness of the proposed system

Multi-Class Classification

Using the chosen MobileNet model it was possible to do IC on more than two classes. In particular, the 4 classes of the main dataset were considered separately. The framework used for this purpose was identical to the one previously described. The best way to visually understand the performances for this task is to plot the confusion matrix in Figure 5.

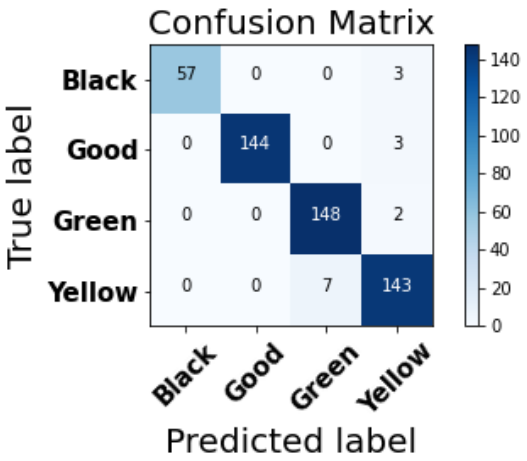

Figure 5 - Result of multiclass classification

It can be seen how the algorithm was capable of classifying both good tomatoes and the ones with anthracnose. On the other hand, some mistakes were made when identifying green tomatoes as well as yellow ones. Still, even without optimizing the algorithm the results were good.

6. Conclusions

The most important results emerging from this work are reported. For the industrial task analysed both AD and IC algorithms reached satisfactory results working on simple RGB images. In fact, all models were able to divide faulty images from good ones with an average accuracy on a 5-fold cross validation ranging from 83.28% for the worst model to 99.14% of the best one. This suggests the possibility to substitute cumbersome human quality control with automated systems.

Overall performances of IC algorithms are superior to the ones of AD algorithms for the task analysed. IC algorithms are both faster and better in terms of AUROC and accuracy. This was concluded with significance tests with a confidence level of 95%. Therefore, the results are statistically significant. For the industrial application studied the best solution is to implement an IC-based model. For the needs of Raytec Vision S.P.A., the best model is MobileNet. It operates with an accuracy of 98.61% and has an inference time of 1.4 ms per image with a realistic GPU setting. With this inference time it can be integrated in machinery like conveyors and sorting hardware.

The here found results could also be interesting for many similar industrial cases that could be automated using one of the presented models. To this end it must be remembered that Anomaly Detection algorithms operate also in cases in which a robust dataset of anomalous sample is difficult or even impossible to achieve. In such situations the here found performances of the best AD model could be captivating. In fact, PatchCore achieves an average accuracy of 94.47 % on a 5-fold cross validation. The average AUROC is of 98.04% and the average inference time of 16 ms per image.

IC algorithms can also be used in numerous similar industrial applications in which the need is not only to divide the object in good and bad samples, but also multiclass IC is requested. This is demonstrated by trying to identify four different tomato categories in the dataset (good, tomatoes with anthracnose, green, yellow) with the chosen MobileNet model and obtaining a 97.04% accuracy.

In future research on this topic, further focus could be set on using the information obtained from the multiclass IC results. It would help in understanding which family of faults creates the biggest problems for the system. Knowing this it is possible to modify the architecture so to be more effective on those specific cases.

Increasing the AD's models speed could be beneficial. This could for example be achieved by implementing them in C++. This is useful since with the here achieved inference times it is not possible to integrate those system with already existing sorting hardware.

It is even possible to reuse the type of pipeline presented above to implement newer models which may rise in the future. Both IC and AD are active fields of research nowadays in the deep learning field. New architecture and models are being proposed so that the proposed models could be surpassed by newer ones in a matter of months.

Using proprietary software which may exhibit some kind of optimized performance with respect to the here presented open-source versions is another field of further analysis.

7. Bibliography

[1] V. Lakshmanan, M. Görner, and R. Gillard, "Practical Machine Learning for Computer Vision End-to-End Machine Learning for Images."

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," 2016. [Online]. Available: http://image-net.org/challenges/LSVRC/2015/

[3] M. Huh, P. Agrawal, and A. A. Efros, "What makes ImageNet good for transfer learning?," Aug. 2016, [Online]. Available: http://arxiv.org/abs/1608.08614

[4] K.-H. Yuan and K. Hayashi, "Bootstrap approach to inference and power analysis based on three test statistics for covariance structure models," 2003. [Online]. Available: www.bps.org.uk

[5] R. Dror, S. Shlomov, and R. Reichart, "Deep Dominance-How to Properly Compare Deep Neural Models," Association for Computational Linguistics. [Online]. Available: https://github.com/

[6] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra, "Grad-CAM: Why did you say that?," Nov. 2016, [Online]. Available: http://arxiv.org/abs/1611.07450

8. Acknowledgements

Authors would like to thank Raytec Vision S.P.A. for providing the datasets used for the experiments and for the insights provided during the project execution.