A Multitask Approach for Type and Quality Fruit Classification with DANN

Domenico Carlucci
s254366@studenti.polito.it

Angela Guastamacchia
s255394@studenti.polito.it

Antonio Paolo Passaro
s246814@studenti.polito.it

Politecnico di Torino
Corso Duca degli Abruzzi, 24, 10129 Torino, Italy

Abstract

Nowadays, computer vision systems have been exploited in multiple fields to ease mechanical and iterative jobs. In particular, they have been associated with harvesting robots to make the fruit selection cost-effective and efficient. So far, two main kinds of systems have been developed: one aimed at recognizing the type of fruit and another focusing only on the maturity analysis of a particular kind of fruit. It could be useful to implement a more versatile system capable of classifying fruit according to its category and, at the same time, flagging spoiled fruits belonging to any category. In addition, this new kind of architecture could be recommended to supermarkets in order to speed up the task of fruit sorting, directly discarding the rotten elements. Therefore, in this perspective, a multitasking CNN has been developed, having two classification branches: a multi-class label predictor to identify the kind of fruit and a binary discriminator to distinguish between fresh and rotten fruits. For this purpose, a new ad hoc dataset has been built up consisting of 20889 clean images and 10321 realistic photos, both representing fresh and spoiled fruits for different categories. To achieve the best possible prediction results, the transfer learning technique has been applied during the model training, performing fine-tuning on uploaded pre-trained models. The proposed framework achieved 99.85% and 99.83% of mean values for the prediction accuracy on clean images for the fruit classification and for the identification of spoiled fruits,respectively. Then, to optimize the classifier behaviour in view of its use in the real world, the framework has been further extended following the DANN one, so as to try reaching high accuracy results also on real images, in practise 68.59% for the fruit classification task and tot 82.87% for the rottenness flagging.

1. Introduction

The introduction of harvesting robots in the industries and other fields, for classifying and analysing fruits in real time, can increase the quality and the time needed for the selection process and the quality inspection. In addition, having an accurate automated vision system that is able to automatically recognize the type of fruit and the quality, at the same time, is very useful and innovative also for supermarkets and wholesalers, allowing to overcome the errors and the manual operators problems due to stress and other issues [2].Although different studies and projects have been conducted in this field, mainly focused on a simple fruits classification task (among different kinds of fruits [4] or among different varieties of the same fruit [2]) or on the quality recognition of a single specific fruit [1], none of them combined both the tasks together. In particular, the work by Y. Zhang et al. [4] shows an example of fruit classifier. It is based on a customized 13-layers CNN which is used to classify 18 different types of fruits, without exploiting transfer learning. They started with an initial small dataset containing 3600 Office31-like images on which they performed data augmentation, in order to obtain in the whole a training set of 63000 images. The overall test accuracy obtained is 94.94% on clean images, while on samples coming from real life conditions it decreases up to 89.60%. Whilst, a quality discriminator is presented in the paper by C. Enciso-Aragón et al. [1]. It focuses only on the quality analysis of one type of fruit, the Persian lemon, and it is composed of two cascaded blocks. The first exploits the AlexNet architecture as CNN, with transfer learning, to operate a preliminary selection of the lemons, since it separates the spoiled lemons from the fresh ones, so that only the fresh lemons are analysed by the second block, through a fuzzy logic, which defines the final lemon quality (according to some predefined characteristics): high, medium or low. Here the dataset is very small and is composed of 320 images. The maximum validation accuracy reached with this architecture is around 97%. Nevertheless, the test accuracy evaluated on 60 images only for the CNN (freshness discriminator) achieves 100%. Ultimately, the work by H.Altaheri et al. [2] implements a more complex analysis only on the dates. In fact, the built architecture takes as images photos of dates and returns three outputs: the variety to which the dates belong to, their stage of maturity and the relative harvesting decision (whether they are ready for harvest or not). The first two decisions are generated by two different CNN, being two simple multiclass classification tasks, while the last output is provided by a binary classifier that makes a decision on the basis of the first two classifiers outputs and on the rules that are manually entered by an external user. Concerning the type and the maturity classifications, this paper investigates the performances obtained through both a pretrained AlexNet and a pretrained VGGNet CNN. In this case the dataset contains 8072 orchard images splitted in 4530 for training and 3542 images for testing. At the end, for the fruit type classifier an accuracy of 96.51% is obtained exploiting the AlexNet architecture while an accuracy of 99.01% is achieved through the VGGNet. Taking a cue from the mentioned works, the idea behind this project resembles the last mentioned paper [2] and is based on the implementation of a more complete classifier, capable of recognizing different kinds of fruits (as in paper [4]) and able to abstract the rottenness feature regardless of the kind of fruit under analysis, so generalizing to multiple types of fruit the rotten/fresh fruit binary classification proposed in paper [1]. However, in general, automatic fruits classification and quality inspection in different contexts are very challenging machine vision tasks due to the different fruits properties (colour, size, shape,..), that have to be taken into account when analysing the images, and due to the different visual appearance changes in the images coming from different context domains [2]. To overcome these aspects, a deep-learning vision system has been exploited. The proposed architecture is a multitask CNN implemented starting from the Alexnet structure which has been modified to execute more than one task. Since it was not feasible to create a dataset containing a large set of labelled target images (realistic images) intended for the training process, the study has been developed in two phases. At first the architecture has been trained exploiting the large amount of available annotated clean images (source data) and subsequently a domain adaptation technique has been implemented in order to try to make the obtained network working well also on target images. Therefore, after having designed the whole classifier structure, it has been handled and trained differently depending on which phase of the classifier study was being faced. In particular, the model is made by a first part corresponding to the Alexnet convolutional layers blocks to which three different parallel classification branches are attached. The first branch focuses on the classification of the type of fruits, the second is a binary classifier that determines the quality of the fruits (fresh or rotten) and the last branch (used only in the second phase) is the domain classifier used to implement the domain adaptation between clean and realistic images (following the DANN algorithm), in order to adapt the built architecture to real world applications and improve the performances on target images. Furthermore, since, for sake of simplicity, the number of recognizable fruits has been limited up to 6 different classes, a further category ("other") has been added. It represents a miscellaneous of different types of fruits that are not present as stand alone categories and it is aimed at making the proposed classifier exploitable for as many applications as possible, so as to have a valid output even when the fruit input to the classifier does not belong to any of the proposed classes. A future enhancement might be to extend the classification to more fruits belonging to different categories and improve the binary fresh/rotten discriminator in a more accurate system for the evaluation of the quality or the maturity stage of the fruit.

2. Multitask CNN for fruit classification

The vision system has been built referring to Convolutional Neural Networks (CNNs), which nowadays represent one of the main employed solutions for object detection, recognition and classification problems. Indeed, thanks to a new feature extraction process (which is directly embedded in the model), they succeed in overcoming the accuracies resulting from the other machine learning techniques [2]. In practise, they are composed of a cascade of layers, falling into two main blocks: the feature extractor and the label predictor. The first is composed of a sequence of hierarchically organized convolutional layers, interleaved with nonlinear functions and sub-sampling layers, which is in charge of learning complex features, seen as compositions of simpler features, directly from the training data input to the network. The returned output, bringing information about the most significant features, is input to the second block made by a series of fully connected layers, aimed at actually implementing the visual recognition task. The training is an iterative process which consists in evaluating a loss function at the network output, by comparing the predictions obtained for the training samples with their actual labels, and finding the parameters values, associated to each layer of the model, which minimize the final loss. As mentioned, the proposed architecture is a multitask CNN, meaning that it has two parallel fully connected blocks, assigned to two different classifications tasks and linked to the same feature extractor, that are trained at the same time. In this case, for both the outputs the respective loss functions are evaluated and summed together so that the minimization is performed on this last loss. In this way, the features extractor is trained in such a way to learn at the same time for the same image the features which distinguish the kind of fruit and the ones which identify the quality (rottenness) of that fruit. However, in order to effectively work, the CNNs need a very large amount of training data and involve high computing efforts, leading to very high training times. Therefore, in order to face this issue, the transfer learning approach has been adopted in this design. In practise, the parameters of the used CNN have been initialized, before the beginning of the training procedure, with values learnt through the training on a large-scale dataset, that is ImageNet, so that the actual performed training consisted only in a fine-tuning of the starting weights based on the actual classification task that the CNN had to implement. In this way, low- and mid-level features are already learnt from the ImageNet dataset, making the training time shorter and the predictions more accurate. Furthermore, the basic CNN architecture actually chosen to accomplish the classification task is the AlexNet model, while the cross entropy function has been chosen as the loss evaluation criterion.

Moreover, in order to implement the domain adaptation, as mentioned in Section 1, the DANN algorithm is implemented in the built architecture. Goal of domain adaptation is to limit the decrease of the performance of visual recognition systems on the test data due to the fact that the training data and the test data belong to different distributions. As in this case, the training data are clean images (all having white background) while the test data come from real scenarios, but both have the same list of categories. Since the problem lies in the fact that the features extracted from the source images (training ones) are far from the ones of the target images (test ones) when the analysed image represents the same object, the idea of the DANN is to force the source features and the target ones to overlap, so that it is no longer possible to distinguish between objects of the target domain and objects of the source one. This is done by adding another parallel binary classifier, in the multitask structure, in charge of distinguishing the target images from the source one during training. Then, during the optimization procedure of the final loss function, the weights of the layers belonging to the feature extractor are updated in such a way to confuse the features of the different domains, thanks to the introduction of a gradient reversal layer [3]. Therefore, the feature extractor generates domain invariant features, being trained in an adversarial manner with respect to the domain binary classifier that inversely tries to distinguish the different domains. In this way, it is possible to exploit the same target data, that are available only for testing, in an unlabelled way also during training to make the final classifier work well on the target data provided with ground truth. The whole structure of the designed classifier is shown in Figure 1.

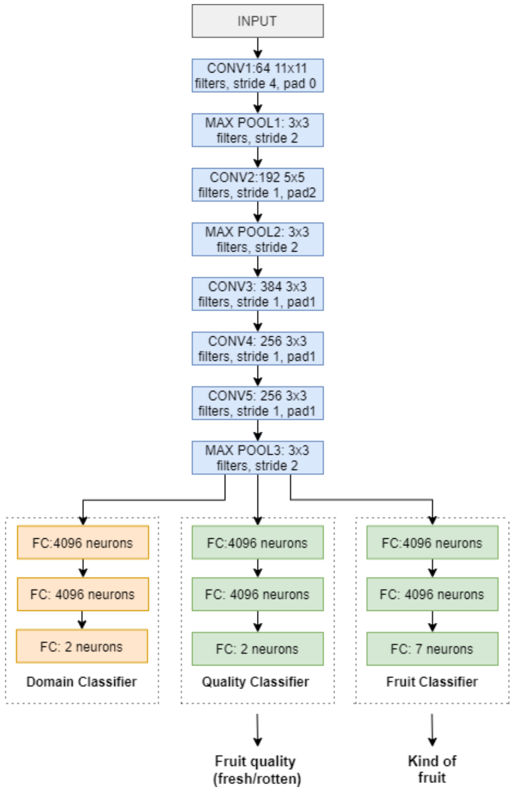
INPUT
CONV1:64 11x11 filters, stride 4, pad 0
MAX POOL1: 3x3 filters, stride 2
CONV2:192 5x5 filters, stride 1, pad2
MAX POOL2: 3x3 filters, stride 2
CONV3: 384 3x3 filters, stride 1, pad1
CONV4: 256 3x3 filters, stride 1, pad1
CONV5: 256 3x3 filters, stride 1, pad1
MAX POOL3: 3x3 filters, stride 2
FC:4096 neurons
FC: 4096 neurons
FC: 2 neurons
Domain Classifier
FC:4096 neurons
FC: 4096 neurons
FC: 2 neurons
Quality Classifier
FC:4096 neurons
FC: 4096 neurons
FC: 7 neurons
Fruit Classifier
Fruit quality (fresh/rotten)
Kind of fruit

Figure 1. Diagram representing the proposed classifier structure.

3. Dataset

During the research for a dataset feasible to perform the training of the CNN, aimed at implementing both the proposed classification tasks, no pre-existing datasets containing suitable realistic images for the fruit quality task have been found. Thus, since the feasible dataset available contained only clean images, in order to implement the domain adaptation, it was necessary to build a new customized dataset, integrating the dataset already found with target images downloaded from the internet. Knowing that a cross validation strategy would have been used for the best model selection, the clean images of the dataset have been preliminarily subdivided into training, validation and test set. While, due to the actual implementation of the training procedure, the realistic images have all been kept in a single set . Therefore the dataset is now available in the github repository (https://github.com/ointona94/Fruits_multitask_classification.git) and composed of four main partitions: train_source, validation_source and test_source (all containing images with white background) and target (images with any kind of background). Each partition contains, separated into fresh and rotten, six categories of fruits (apple, banana, lemon, orange, pear, strawberry) plus an "other" category, as previously mentioned. Train, validation and test contain 10683, 4174 and 6032 images, respectively, that are approximately subdivided according to the proportion 5:2:3. The amount of samples of the target is 10321 that is compatible with the training set size. For each type of fruit there is almost the same amount of images among fresh and rotten categories. The dataset has been made starting from two existing datasets found on internet: *fruits-fresh-and-rotten-for-classification* and *fruit360*, containing only source images. The first one contains train and test, separated into fresh and rotten categories, for only three types of fruits (apple, banana and orange). The second one contains the fresh version of many types of fruits. Both dataset have been enlarged by their own creators using data augmentation. The dataset taken into account for this project has been created starting from a merged version of these two datasets. The building process consisted in a trade-off between the computation time required by the processing of a large dataset and the number of samples required to properly cover each category of fruits: fresh and rotten, for source and target. All the missing categories have been integrated using images retrieved from internet (by means of an ad-hoc built tool aimed at automating both the download of .jpg,.jpeg,.png images and the filtering process), taking further photos (some transformed into the source ones using PhotoShissors as background cleaning software) and extracting frames from video about time-lapse fruits decomposition. At the end of the preprocessing phase all the images had more or less the same dimensions and had the subject well centered in the middle of the image. After a preprocessing phase aimed at balancing the number of acquired images for each specific category (*e.g*. rotten banana) using a python script, the final dataset (neglecting the portion coming from the starting dataset that was already augmented) has been further enlarged by a factor of 24 exploiting the data augmentation technique by applying all the possible combinations of label-preserving transformations: the rotation from 0 to 150 degrees using an anti-clockwise step of 30 degrees and the vertical and horizontal flips. Finally, another script has been used to shuffle and put the images in the respective partitions according to the structure described at the beginning.

4. Experimental results without DANN

In the first phase of this study the model training has been performed only for the first two branches (thus excluding the domain classifier), using, as labelled images, the ones belonging to the source images set assigned for training. This initial setting aimed at retrieving a first estimation for the behaviour of the designed architecture in the simplest conditions. The best hyperparameters search has been carried out looking at the model prediction accuracies retrieved on the validation set composed only of source images. The best values found for the hyperparameters, adopting the SGD with momentum (equal to 0.9) optimization method for the updating of the architecture weights, are: learning rate = 1e-3, weight decay (regularization term) = 5e-5, batch size = 128, maximum epochs number = 40. Moreover, no step-down policy of the learning rate has been used during the training over the epochs. Furthermore, since the batches of training data are composed of shuffled data that change at every independent run, different training procedures have been performed in order to obtain more reliable averaged values for the validation accuracies. In Figure 2 and Figure 3 the averaged results achieved with the optimum hyperparameters during validation are shown for both the type and the quality classifiers. As it can be seen, the model is able to reach high values of accuracy on the validation set for both the tasks, without leading to any generalization gap (no training data overfitting). The two classifiers, whose purpose is to distinguish among different types of fruits and to evaluate the fruit quality (fresh/rot), behave similarly reaching on the source validation set, at the 31th epoch, a mean accuracy of 99.94% with a standard deviation lower than 0.04%. Then, on this best model, the test set containing only source images has been applied, in order to retrieve the final accuracy values: namely 99.85% for the fruit classifier and 99.83% for the quality classifier (also called rottenness classifier). In Table 1 and Table 3) the confusion matrix and the precision, the specificity and the sensitivity values of the fruit classifier on the test test, coming from one run, are reported. Similarly, in Table 2 and Table4, the results for the quality classifier are shown. From the observation of the classifiers performance, shown by the evaluated metrics, it follows that, for both the tasks, most of the made predictions were correct, as specifically confirmed by looking at the values present on the main confusion matrix diagonal, which reach almost the cardinality of the class to which they refer. Afterwards, in order to estimate the behaviour of the designed system on real world applications, the model has been further tested on the target images. As expected, the performances dropped down down for both the fruit kind and the quality classification tasks, reaching accuracies of 65.58% and 80.86%, respectively, highlighting the different domains issues between training and test images.

5. Experimental results with DANN

The second phase saw the attempt to improve the results, obtained on the target images, including in the model training also the domain classifier branch, in order to implement the adaptation between the source and the target domain. In particular, the network has been trained on the annotated source data and on the unlabelled target data (forwarded only to the domain classifier) and after it has ben evaluated on the source validation set. At last, the test phase has been carried out on the same target set, but exploiting the related ground truth. As before, also in this step

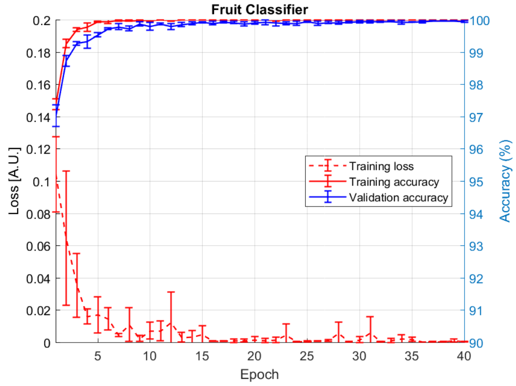

Figure 2. Mean values and respective standard deviations for training loss, training and validation accuracy evaluated on 3 runnings across 40 epochs on the fruit classifier.

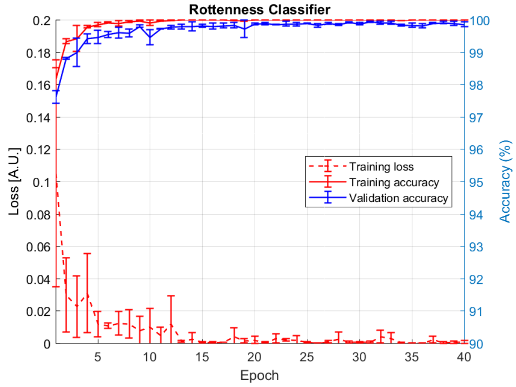

Figure 3. Mean values and respective standard deviations for training loss, training and validation accuracy evaluated on 3 runnings across 40 epochs on the quality classifier.

class	1	2	3	4	5	6	7
1	901	0	0	1	0	0	0
2	0	897	1	0	0	1	0
3	0	0	900	0	0	0	0
4	2	0	0	900	0	0	0
5	1	0	0	1	746	1	0
6	0	0	0	1	0	791	0
7.	0	0	0	0	0	0	888

Table 1. Confusion matrix for the fruit classifier resulting from test on source data. Classes association: 1→apple, 2→banana, 3→lemon, 4→orange, 5→other, 6→pear, 7→strawberry

class	fresh	rotten
fresh	3091	3
rotten	7	2931

Table 2. Confusion matrix for the rottenness classifier resulting from test on source data.

class	precision [%]	sensitivity [%]	specificity [%]
apple	99.67	99.89	99.94
banana	100	99.78	100
lemon	99.89	100	99.98
orange	99.67	99.78	99.94
other	100	99.60	100
pear	99.75	99.87	99.96
strawb.	100	100	100

Table 3. Performance metrics for the fruit classifier resulting from test on source data.

class	precision [%]	sensitivity [%]	specificity [%]
fresh	99.77	99.90	99.76
rotten	99.90	99.76	99.90

Table 4. Performance metrics for the rottenness classifier resulting from test on source data.

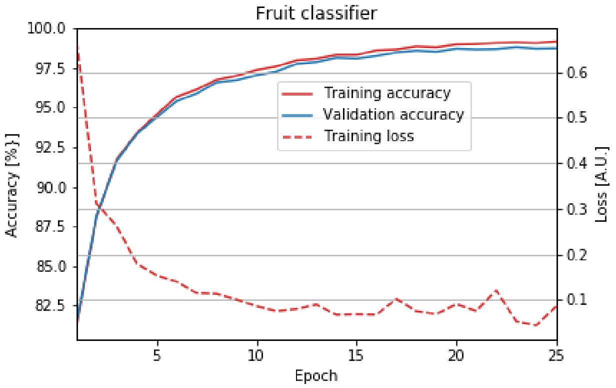

Figure 4. Training loss, training and validation accuracy for the fruit classifier exploiting the DANN.

the hyperparameters tuning has been performed involving also the “alpha” parameter, used in the implementation of the updating procedure of the DANN weights. From a grid search, performed according to a “best validation accuracy policy”, the best hyperparameters found were: learning rate = 1e-4, alpha=0.5, batch size = 256, maximum number of epochs = 25. Again, no step-down policy for the learning rate has been enabled during training. The accuracy values on the validation set for both the tasks are shown in Figure 4 and 5, while Figure 6 collects the training losses of the three classifiers across the epochs. At the 23^{th} epoch the fruit classifier reaches the maximum validation accuracy of 98.8%, while the quality classifier achieves 98.51%. Selected this best model, the test procedure has been run and the following accuracy results have been retrieved: 68.59% for the fruit classifier and 82.87% for the rottenness one. All the evaluated metrics for these classifiers are shown in Tables 5, 6, 7 and 8. By comparing the obtained results with the ones retrieved for the target in the previous phase, it follows that using DANN helps increasing the overall accuracy only by 3 percentage points for the fruit classification task, passing from 65.58% to 68.59%, and of about 2 percentage point, from 80.86% to 82.87%, for the quality clas-

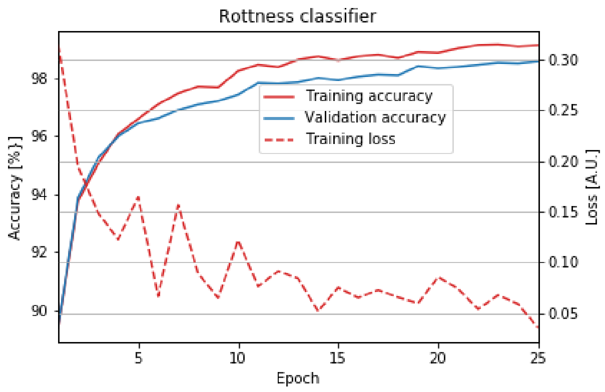

Figure 5. Training loss, training and validation accuracy for the rottenness classifier exploiting the DANN.

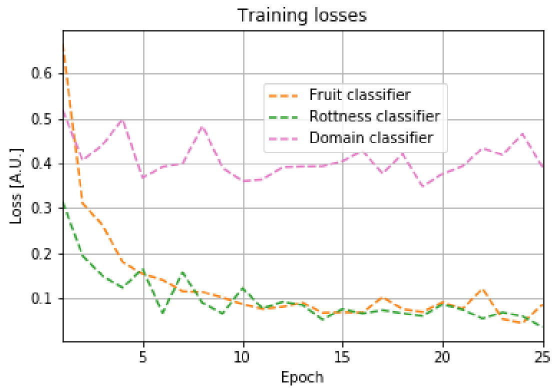

Figure 6. Training losses for the fruit, the quality and the domain classifiers.

class	1	2	3	4	5	6	7
1	802	0	157	30	328	129	48
2	20	1284	0	0	32	103	1
3	19	9	1012	53	103	299	5
4	12	0	633	709	26	23	67
5	22	36	49	99	1127	108	30
6	145	33	64	206	252	768	32
7	0	0	0	6	63	0	1377

Table 5. Confusion matrix for the fruit classifier resulting from testing on target data with DANN. Classes association: 1→apple, 2→banana, 3→lemon, 4→orange, 5→other, 6→pear, 7→strawberry

sification task. Thus, the DANN implementation does not work as expected. In fact, looking at the loss of the domain discriminator, it is clear that the model is still able to correctly distinguish the two different domains, meaning that it does not succeed in overlapping the source features with the target ones, leading to low prediction accuracy values for the target images. This may be due to the too big difference in the visual appearances between images belonging to the target domain and images coming from the source one.

class	precision [%]	sensitivity [%]	specificity [%]
apple	78.63	53.68	97.53
banana	94.28	89.17	99.12
lemon	52.85	67.47	89.76
orange	64.28	48.23	95.55
other	58.36	76.61	90.92
pear	53.71	51.20	92.50
strawberry	88.27	95.23	97.94

Table 6. Performance metrics for the fruit classifier resulting from testing on target data with DANN.

class	fresh	rotten
fresh	4012	1124
rotten	644	4541

Table 7. Confusion matrix for the rottenness classifier resulting from testing on target data with DANN.

class	precision [%]	sensitivity [%]	specificity [%]
fresh	86.17	78.12	87.58
rottenness	80.16	87.58	78.12

Table 8. Performance metrics for the rottenness classifier resulting from testing on target data with DANN.

Moreover, as can be noticed from the confusion matrix (see Table 5), the most significant misclassifications occur for three type of fruits: about 41% of apples are significantly confused with lemons, other and pears, while about 43% of oranges are confused with lemons and finally about 40.2% of pears are confused with apples, oranges and other. In general, this behaviour could be due to the poor quality of the samples available in the target dataset. It is possible, in fact, that the presence of images representing a too rotten fruit does not help the model to distinguish the type of fruit, as well as the presence of occluded fruits placed in a too elaborated environment representing the background. Furthermore, since the dataset has been created doing a trade-off aimed at guaranteeing a sufficient number of samples per class and a sufficient image quality, possible reasons for this poor model behaviour could be the presence in the target dataset of less representative images for some type of fruits, or simply the presence of images belonging to multiple different domains.
So, for future enhancements, it could be useful to reanalyse the portion of the dataset containing the target samples, substituting the poor images with better ones, in order to achieve on the target images the same accuracy reached by the proposed model on source ones.

References

[1] Robinson Jimenez-Moreno Carlos Javier Enciso-Aragón, César Giovany Pachón-Suescún. Quality control system by means of cnn and fuzzy systems. *International Journal of Applied Engineering Research*, 13(16):12846–12853, 2018.

[2] Ghulam Muhammad Hamdi Altaheri, Mansour Alsulaiman. Date fruit classification for robotic harvesting in a natural environment using deep learning. *IEEE Access*, 7:117115–117133, 2019.

[3] Hana Ajakan Pascal Germain Hugo Larochelle-François Laviolette Mario Marchand-Victor Lempitsky Yaroslav Ganin, Evgeniya Ustinova. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17:1–35, 2016.

[4] Xianqing Chen Wenjuan Jia Sidan Du Khan Muhammad Shui-Hua Wang Yu-Dong Zhang, Zhengchao Dong. Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation. *Multimedia Tools and Applications*, 78(3):3613–3632, 2019.