How efficient deep-learning object detectors are?

Luis Miguel Soria [a, *], Francisco J Ortega [b], Juan A. Álvarez-García [a], Francisco Velasco [b], Damián Fernández-Cerero [a]

[a] Departamento de Lenguajes y Sistemas InformÃticos, Universidad de Sevilla, Spain
[b] Departamento de EconomÃa Aplicada I, Universidad de Sevilla, Spain

Keywords:
Neural networks
Deep learning
Object detection Efficiency analysis
Data envelopment analysis

A B S T R A C T

Deep-learning object-detection architectures are gaining attraction, as they are used for critical tasks in relevant environments such as health, self-driving, industry, security, and robots. Notwithstanding, the available architectures provide variable performance results depending on the scenario under consideration. Challenges are usually used to evaluate such performance only in terms of accuracy. In this work, instead of proposing a new architecture, we overcome the limitations of those challenges by proposing a computationally undemanding comparative model based on several Data Envelopment Analysis (DEA) strategies, not only for the comparison of deep-learning architectures, but also to detect which parameters are the most relevant features for achieving efficiency. In addition, the proposed model provides with a set of recommendations to improve object-detection frameworks. Those measures may be applied in future high-performance meta-architectures, since this model requires lower computational and temporal requirements compared to the traditional strategy based on training neural networks – based on the trial-error method – for each configurable parameter. To this aim, the presented model evaluates 16 parameters of 139 configurations of well-known detectors present in the Google data set [1].

1. Introduction

Object detection is a research area that is attracting interest in recent years. Deep-learning architectures are being used in diverse object-detection contexts [2], such as: (a) computer vision [3,4]; (b) robots [5]; and (c) health [6]. The performance of the deep-learning architectures on each context is sometimes hard to evaluate, being such analysis crucial to determine which architecture is the best suited for each context.

Moreover, new architectures, feature extractors, and combinations of both of them are continuously presented. A fair comparison between different object detectors is also difficult to carry out due to the high number of characteristics under analysis. Such architectures and detectors are commonly evaluated by challenges, which consider common objects, such as pedestrians, and vehicles. These challenges include: Imagenet [7], PASCAL VOC [8], and Microsoft COCO [9]. In these challenges, only speed and accuracy are considered. Therefore, several other relevant metrics may be left out of the evaluation process.

In this paper, we overcome this limitation by employing Data Envelopment Analysis (DEA) for the comparison of several well-known detectors based on SSD, R-CNN, and R-FCN. Since DEA is claimed to lack homogeneity between different models, we employ three different DEA models to improve the robustness and flexibility of the analysis performed: (a) the original DEA model (managerial); (b) a DEA model based on probabilistic techniques; and (c) a DEA model based on Bayesian techniques. The main aims of the experimentation performed are: (a) to provide a formal framework to determine which combinations of detectors, as well as their related input and output parameters, are efficient (related to performance in this work); (b) to empirically detect which parameters affect the most to the performance of object-detection frameworks; (c) to provide a model able to formally propose a set of in-detail corrections to improve each parameter of the analysed object detection models.

The main contributions of this work include:

1. A model for the efficiency analysis of different neural-network configurations.
2. A formal framework for the determination of the most crucial parameters for object detection performance; and
3. A simple and low-computing-demanding methodology to improve the efficiency for deep-learning architectures.

* Corresponding author.
E-mail addresses: lsoria@us.es (L.M. Soria), fjortega@us.es (F.J. Ortega), jaalvarez@us.es (J.A. Álvarez-García), velasco@us.es (F. Velasco), damiancerero@us.es (D. Fernández-Cerero).

Table 1
DEA inputs and outputs. Action column arrows mean whether the input/output value may be decreased (down arrow), increased (up arrow), or kept equal. This makes reference to positive and negative inputs/outputs.

Parameter	Description	Action
Inputs		
Image resolution	Size of the input images of the training set	↑ ↔
Proposals sent to box classifier	Number of regions of the original image sent to the detection classifier	↑ ↔
Outputs		
mAP	Global mAP (accuracy) for the whole set of images in the validation set	↑ ↔
mAP (small)	mAP for small-sized region of interest	↑ ↔
mAP (medium)	mAP for medium-sized region of interest	↑ ↔
mAP (large)	mAP for large-sized region of interest	↑ ↔
mAP @. 75IOU	mAP with an 75% overlapping threshold between ground truth and bounding box	↑ ↔
mAP @. 50IOU	mAP with an 50% overlapping threshold between ground truth and bounding box	↑ ↔
Memory	RAM memory consumed during learning process (MiB)	↓ ↔
Memory std	Average deviation of memory consumption (MiB)	↓ ↔
CPU	CPU time consumed by each batch (milliseconds)	↓ ↔
CPU std	Time deviation of CPU consumption (milliseconds)	↓ ↔
GPU std	Time and deviation of GPU utilisation	↓ ↔
FLOPS	Floating operations during training	↓ ↔

Table 2
Sample of the data set used for DEA analysis.

#	Architecture, extractor	Resolution	mAP	mAP @. 75IOU	# params	Memory	CPU	GPU	FLOPS
1	Faster RCNN, VGG	300	22.90	22.90	138.51	1699.00	27.00	16.00	64.32
2	Faster RCNN, VGG	300	19.60	20.40	138.51	1703.00	41.00	16.00	29.12
11	Faster RCNN, Resnet 101	300	26.50	27.20	63.15	3236.00	28.00	14.00	239.42
18	Faster RCNN, Resnet 101	600	29.40	31.80	63.15	1985.00	93.00	16.00	91.35
31	Faster RCNN, Inception V2	300	15.40	14.80	13.31	2370.00	5.00	15.00	118.22
32	Faster RCNN, Inception V2	300	13.30	13.40	13.31	211.00	7.00	13.00	7.63
49	Faster RCNN, Inception V3	600	28.60	29.90	26.27	1522.00	60.00	16.00	74.37
50	Faster RCNN, Inception V3	600	29.20	30.20	26.27	2293.00	5.00	13.00	124.51
51	Faster RCNN, Inception Resnet V2	300	28.40	29.30	60.02	14769.00	104.00	15.00	639.98
70	Faster RCNN, Inception Resnet V2	600	35.30	37.60	60.02	10341.00	38.00	14.00	370.69
71	Faster RCNN, MobileNet	300	16.40	15.50	6.06	1147.00	22.00	17.00	25.23
72	Faster RCNN, MobileNet	300	14.40	14.20	6.06	143.00	5.00	7.00	2.55
81	R-FCN, Resnet 101	300	25.20	25.90	68.48	992.00	15.00	6.00	27.14
82	R-FCN, Resnet 101	300	25.20	26.00	68.48	861.00	25.00	5.00	27.14
101	R-FCN, Inception V2	300	15.40	14.50	18.06	286.00	7.00	5.00	5.09
102	R-FCN, Inception V2	300	15.50	14.60	18.06	208.00	6.00	6.00	5.09
111	R-FCN, Inception Resnet V2	300	22.50	22.70	65.06	696.00	10.00	6.00	14.99
112	R-FCN, Inception Resnet V2	300	22.80	23.00	65.06	642.00	13.00	7.00	14.99
131	R-FCN, MobileNet	300	15.00	13.70	10.80	246.00	4.00	5.00	2.44
132	R-FCN, MobileNet	300	15.20	13.80	10.80	170.00	8.00	7.00	2.44

The rest of this paper is organised as follows. In Section 2, we briefly introduce DEA and object detectors. A formal definition of the employed DEA models is provided in Section 3. The analysis performed and obtained results are outlined in Section 4. Finally, in Section 5, we provide the conclusions and discuss the future work.

2. State of the art

In this section, we discuss the literature regarding DEA and object detection.

2.1. Data envelopment analysis

DEA was suggested in [10] to quantify productive efficiency of decision-making units (DMU). This non-parametric method assesses efficiency through the analysis of a set of inputs and outputs. DEA has been proved useful to empirically compute the efficiency of DMUs and to determine whether such DMUs belong or not to the "frontier" of production. In this context, the frontier of production, that is, the efficiency frontier, is composed of the most efficient DMUs, and is utilised to compute the relative efficiency of the rest of DMUs. DEA is a practical tool when diverse inputs and outputs with unknown relationships between them are under consideration. Although DEA was initially employed only in operations research and economic environments, several applications of DEA have emerged for the measurement of the efficiency in diverse areas, such as:

(a) energies [11,12]; (b) decision-support systems [13,14]; (c) research evaluation [15–17]; (d) public sector analysis [18–20]; (e) entrepreneurship [21]; (f) environment [22,23]; and (g) agriculture [24–26].

One of the main advantages of DEA is that it sets up a best-practice production frontier that may be used to decide the level of inefficiency of DMUs. There are several possible DEA modes:

- The original approach proposed in [10] is input-oriented. Therefore, the desired output level is obtained by minimising the input production, and assumes constant returns to scale (CRS): the increment of inputs lead to an increase of outputs.
- Another extended model, proposed in [27], is output-oriented. In this case, DEA tries to maximise outputs with a set of fixed available inputs.
- Finally, the model proposed in [28] employs variable returns to scale (VRS). In this model, an increment of inputs may induce a change in the amount of outputs generated [29].

2.2. Object detectors

Detection architectures are constantly evolving, improving metrics for the detection of various kind of images, such as: (a) faces [30]; (b) handguns [31]; and (c) traffic signs [32]. State-of-the-art object-detection algorithms, such as R-CNN, R-FCN, SSD, and YOLO [33–36], use deep-neural-network architectures. There are several metrics (outputs) to take into account to correctly measure the efficiency of each algorithm in various situations, including: (a) mean average precision (mAP); (b) inference times; and (c) memory consumption. Such measurement may help to decide which detector suits best a particular application. For instance, mobile devices require lightweight-model architectures with low memory usage, whilst autonomous vehicles require good detection accuracy and real-time performance.

These detectors have been evaluated in several object-detection competitions, such as: (a) Imagenet [7]; (b) PASCAL VOC [8]; and (c) Microsoft COCO [9]. There exist comparisons of these detectors in specific contexts [37]. Notwithstanding, the long computation time needed to train them with several configurations (inputs parameters) may prevent the vast majority of users from evaluating them, leaving large companies, such as Google, as the only actor to analyse them [1]. As a result of such analysis, meta-architectures of Faster R-CNN, R-FCN, and SSD have been proposed, and have been combined with various feature extractors, such as: (a) VGG [38]; (b) Resnet 101 [39]; (c) Inception V2 [40]; (d) Inception V3 [41]; (e) Inception Resnet V2 [42]; and (f) Mobile Net [43], in order to compare a large number of detection systems in a unified manner.

In this work, we extend the state of the art by proposing a formal model for efficiency evaluation of deep-learning architectures, as well as providing a systematic framework for the improvement of efficiency of deep-learning architectures based on several relevant parameters.

3. DEA models

DEA assumes data to be free of measurement errors and is sensitive to outliers. Numerous works that introduce stochastic variation in production relationships and provide statistical inference in DEA contexts have been proposed recently [44]. In this work, we employ first the chance-constrained programming technique [45,46]. Assuming that the inputs and the outputs observed are drawn from a joint probability distribution which parameters are unknown, this approach allows exceptional cases with constraint violation (concretely, with a probability lesser than a fixed value α). In a second stage, we take prior distributions for the unknown parameters, and we apply Bayesian Inference to obtain statistical measurements of DMUs efficiency.

Appendix A shows a detailed explanation of the mathematical development for each of the three DEA models used in this paper: managerial, probabilistic and Bayesian.

4. DEA analysis

In this Section, we present the structure of the data set used to carry out the comparison of the different architectures of neural networks, as well as the result of the process of applying the DEA models on the data set described. The analysis has been carried out by means of three different DEA models in order to test whether the conclusions provided by them are homogeneous or not, as well as the different nuances of each DEA model.

4.1. Data set

In this work, we employ the data set generated in [1], which includes the data and parameters for each detector, summing up 139 different configurations. It should be borne in mind that we do not aim to deeply describe the characteristics of each detector under comparison ([1] does), but rather to perform a statistical multi-criteria DEA analysis for the determination of the relative efficiency of a deep learning system in comparison with other existing systems. This data set is composed by the

Table 3
Summary of the result of the managerial DEA analysis (efficiency coefficients) grouped by image resolution for each of the analysed architectures.

Input image resolution	Average DEA coefficient	Min DEA coefficient	Max DEA coefficient
Faster RCNN			
Inception Resnet V2			
300	0.7229	0.5841	1
600	0.8203	0.6640	0.9926
Inception V2			
300	0.8261	0.7094	0.9525
600	0.8692	0.6974	1
Inception V3			
300	0.7991	0.6229	1
600	0.8852	0.6280	1
MobileNet			
300	0.9507	0.8768	1
600	1	1	1
Resnet 101			
300	0.9317	0.6910	1
600	0.9191	0.7925	1
VGG			
300	0.9975	0.9929	1
600	0.9873	0.9367	1
R-FCN			
Inception Resnet V2			
300	0.9899	0.9315	1
600	0.9831	0.8972	1
Inception V2			
300			
600	0.9905	0.9749	1
MobileNet			
300	1	1	1
600	1	1	1
Resnet 101			
300	0.9731	0.8850	1
600	0.8845	0.5912	1

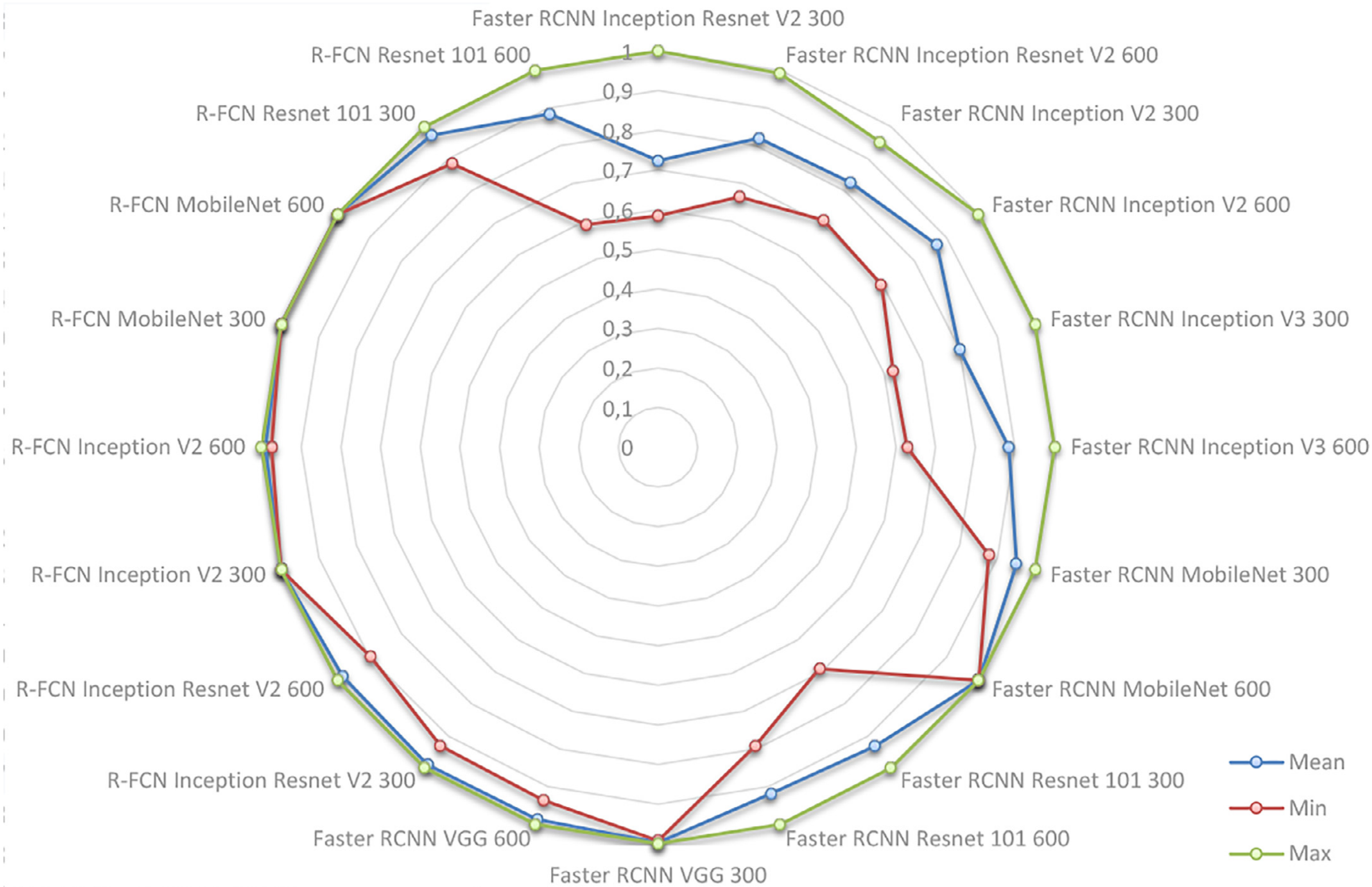

Fig. 1. DEA Managerial method maximum, minimum and average efficiency over different detection techniques based on the resolution of the input images.

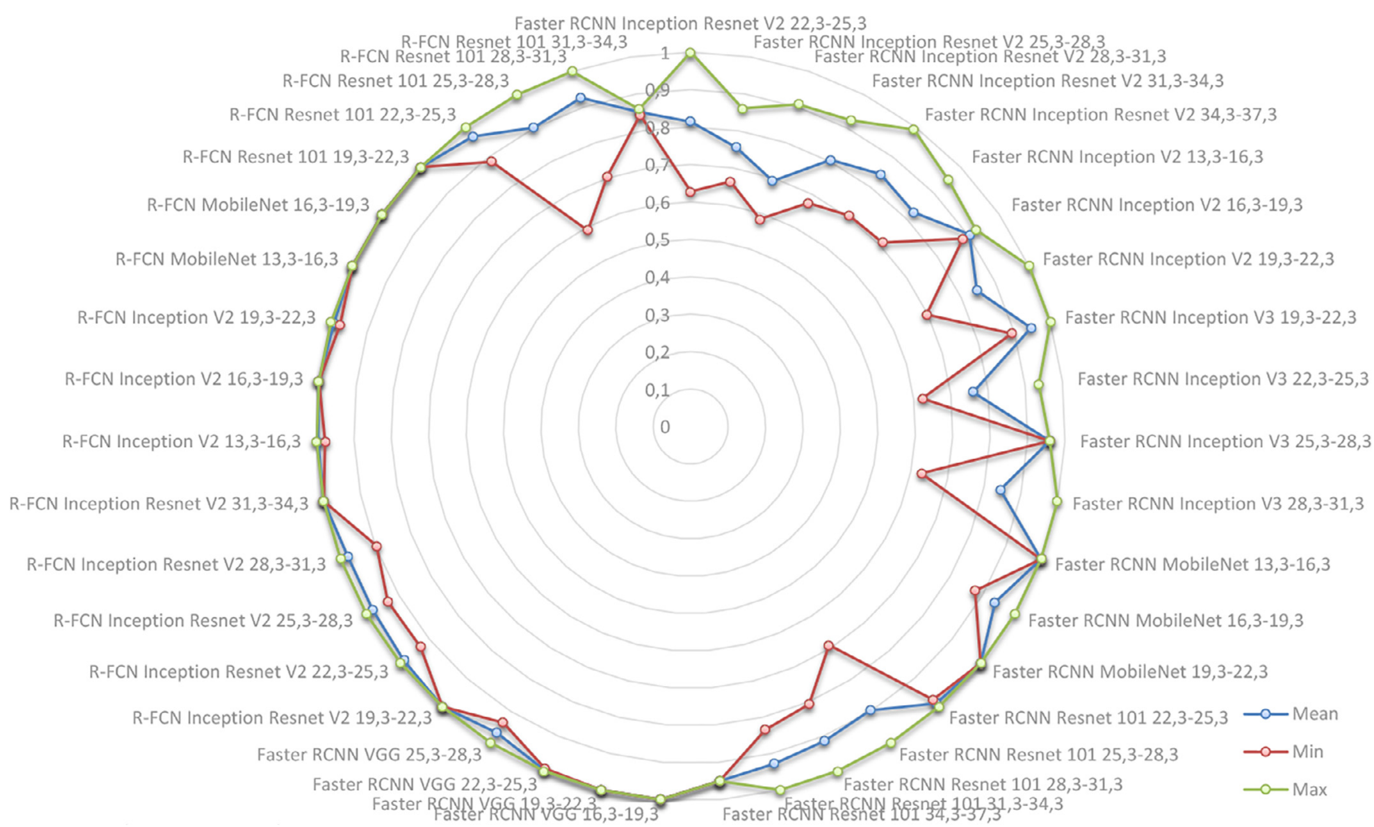

Fig. 2. DEA Managerial method maximum, minimum and average efficiency over different detection techniques based on the box detection (mAP) accuracy.

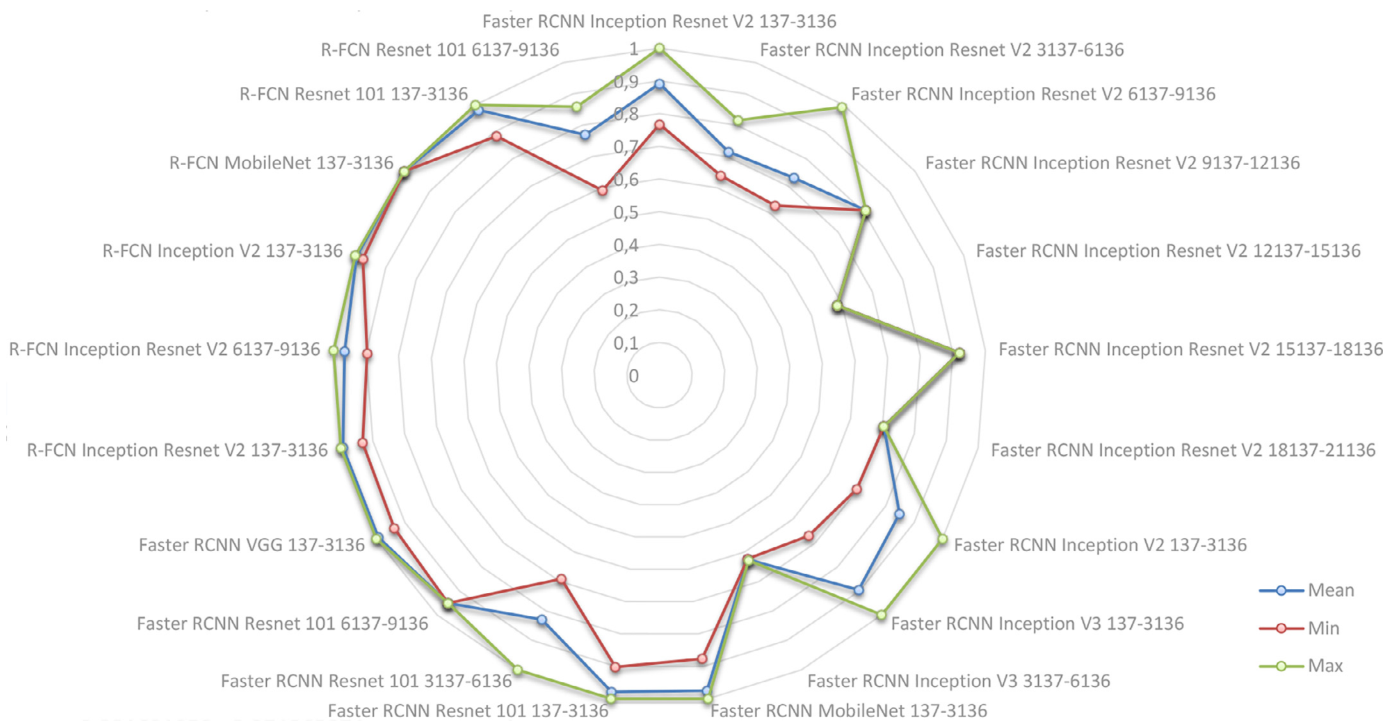

Fig. 3. DEA Managerial method maximum, minimum and average efficiency over different detection techniques based on the consumed memory.

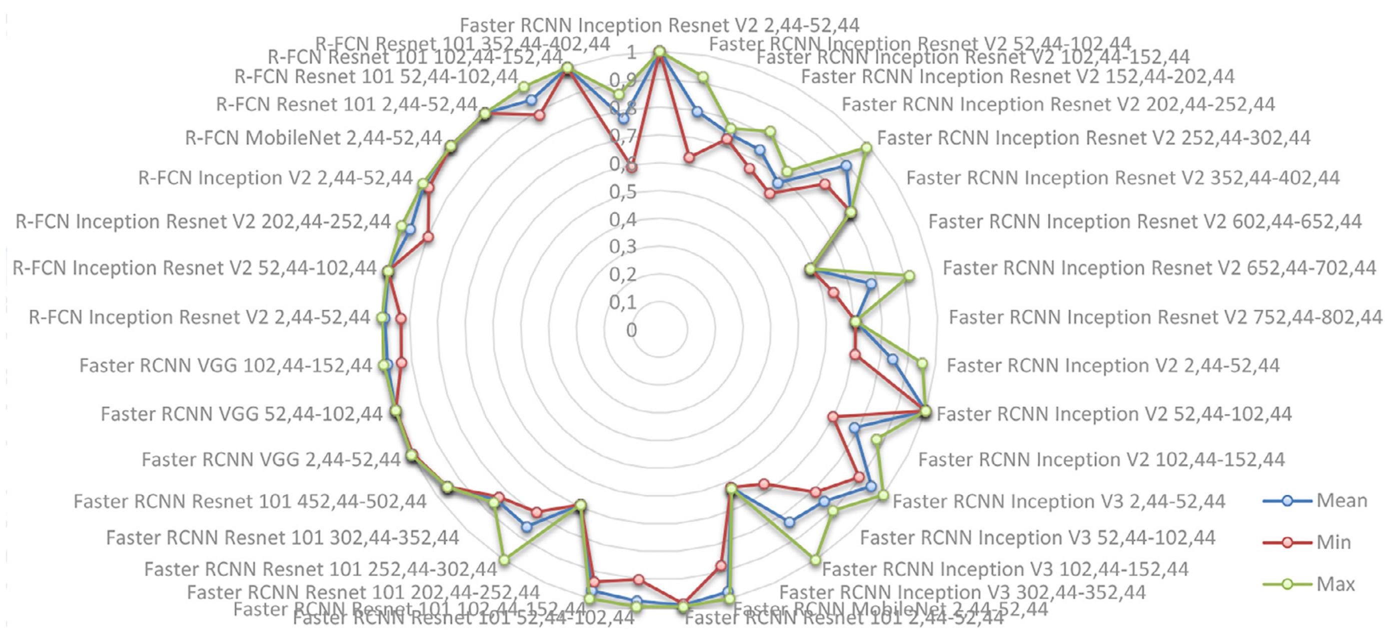

Fig. 4. DEA Managerial method maximum, minimum and average efficiency over different detection techniques based on the number of operations required to complete the whole process.

parameters shown in Table 1, corresponding to the settings of each architecture analysed.

A representative sample of the data set is shown in Table 2. Note that some parameters of the networks have been omitted from Table 2 to increase clarity. Yet, these parameters are described in the next section in order to subsequently compare the efficiency criteria.

The classification as inputs and outputs of these parameters to perform the DEA analysis is presented in the following section.

4.2. DEA inputs and outputs

Once data set has been presented, we discuss how the parameters described in Section 4.1 are taken as inputs or outputs for DEA analysis. Both the number of outputs and inputs may be scaled in order to obtain more information about the efficiency frontier. Nonetheless, due to the temporary cost of the training, these data were reused from previous studies in order to apply the DEA analysis. Moreover, a great advantage of this analysis system is

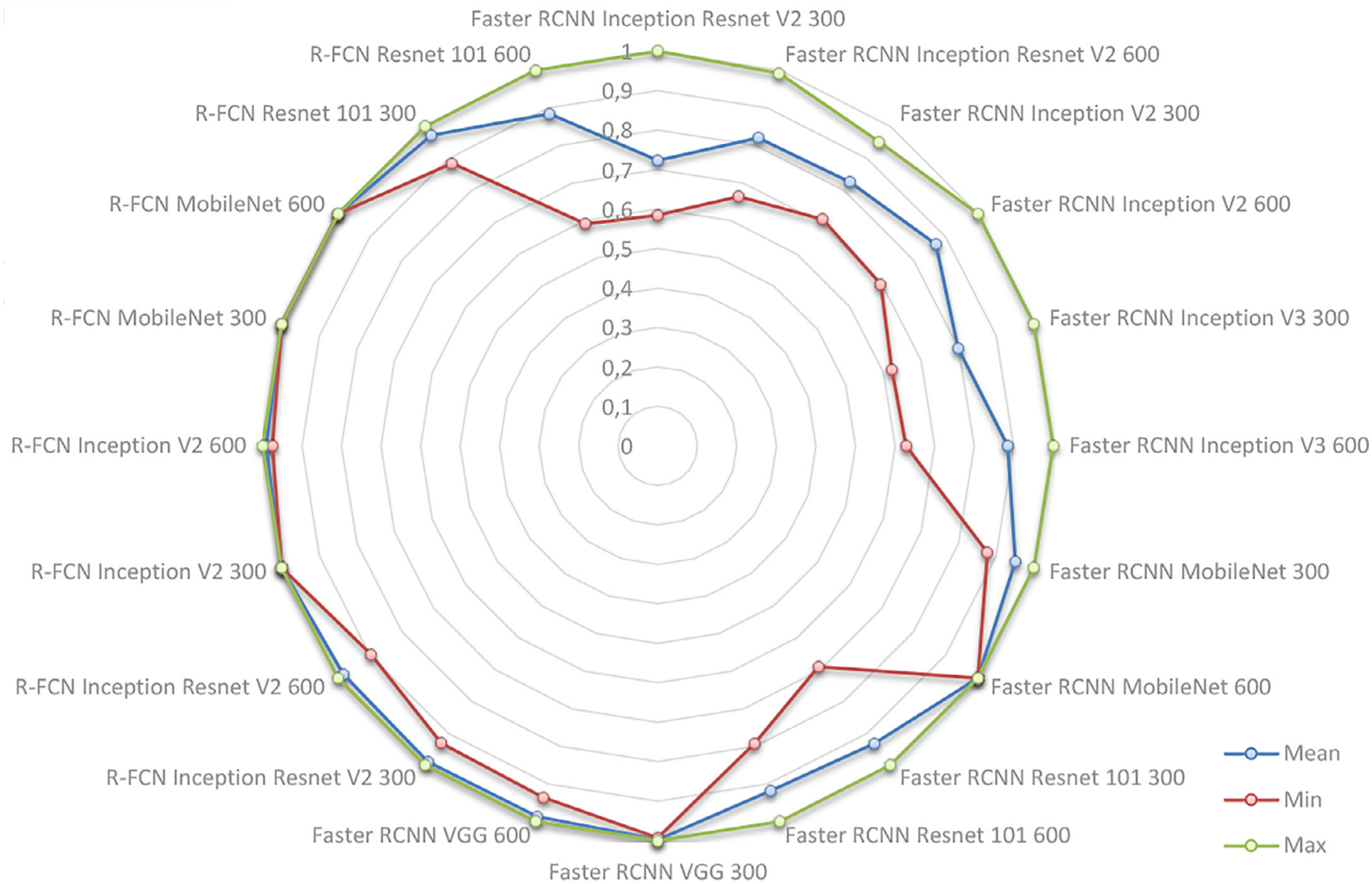

Fig. 5. DEA probabilistic method maximum, minimum and average efficiency over different detection techniques based on the resolution of the input images.

that, due to its relatively low computational cost, it is completely feasible to incorporate new architectures into the system with different configuration values. Therefore, we are able to determine the effect of these new architectures on the efficiency frontier.

The inputs considered in this study are: (a) resolution of the images with which the model works, in pixels; (b) number of regions of interest sent to the classifier for each image (proposals sent to box classifier); and (c) millions of model parameters where (a) and (c) are considered as deterministic inputs. Other parameters such as the number of convolutional layers (RELU, dropout or pooling layers), and the number of features extracted directly from the image have not been analysed.

DEA analysis considers two groups of outputs, as shown in Table 1: positive outputs (good) and negative (bad). The first group refers to the set of outputs that, the higher their value, the better impact they have on the performance of the selected model. On the other hand, the negative outputs represent the set of outputs that, the lower their value, the better impact they have on the performance of the model.

Regarding the positive outputs of the considered models, they are related to the average precision (mAP) with the following different considerations: (i) general mAP for the whole set of images in the validation set; (ii) mAP depending on the size of the region of interest detected (large, medium or small); and (iii) mAP @. 75IOU and mAP @. 50IOU, which refer to the average precision with an overlapping threshold between the ground truth and the bounding box generated of 70% and 50%, respectively.

As for the negative outputs, 7 characteristics of the considered models were taken into account: (i) memory consumption during learning process (MiB); (ii) floating point operations required for training; (iii) CPU time consumed by each batch (milliseconds); (iv) memory consumption during learning process (MiB); (v) average deviation of memory consumption (MiB); (vi) time deviation of CPU consumption (milliseconds); and (vii) time and deviation of GPU utilisation.

Once the data set and the inputs and outputs that make up the input parameters of the DEA analysis have been presented, this analysis has been applied to the previous data. Since this paper addresses 3 different methodologies for the efficiency analysis, the results for each of these methodologies are presented below.

4.3. Managerial DEA analysis

The original managerial DEA analysis was the first model applied to the data described above. This analysis resulted in 69 of the 139 models as belonging to the decision boundary. This means that these 69 models with a maximum efficiency (belong to the efficiency frontier), will be taken as a reference for the projection of inputs and outputs from other non-efficient models.

An example of this process can be seen in the case of the Faster RCNN architecture with Inception V2 with an image resolution of 600 pixels and 300 proposals sent to box classifier (efficiency value of 69,7%), respecting to R-FCN architecture with Resnet 101, with an image resolution of 300 pixels and 20 proposals sent to box classifier (100% efficiency value). It should be borne in mind that this process is a multivariate analysis and is quite difficult to figure out with the naked eye. If outputs of these both units are explored, the average precision of the first falls to 21.9%, while the

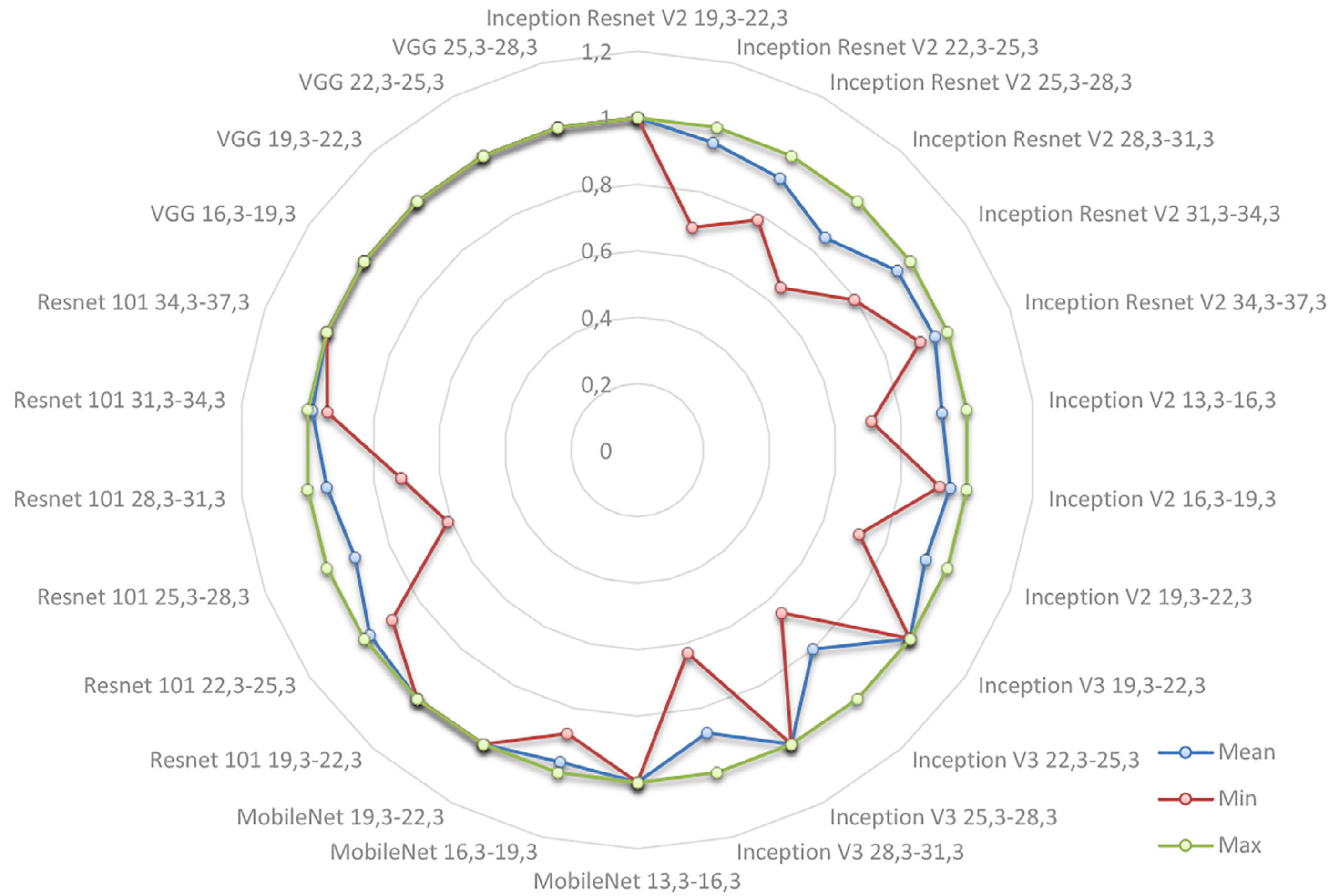

Fig. 6. DEA probabilistic method maximum, minimum and average efficiency over different detection techniques based on the box detection (mAP) accuracy.

second rises to 30.1%. This is also reflected in the average accuracy measured on different image sizes. In contrast, the memory consumption of the Faster RCNN based model is much higher than in the R-FCN. As DEA analysis considers both the outputs and the inputs to produce the efficiency frontier, in this case the increase in detection accuracy counteracts the increase in computational capabilities needed to execute the process of training the model. To better illustrate the comparison of the efficiency values provided by the DEA analysis on the two selected models, an analysis on the accuracy percentiles for both models on the overall result is performed. In R-FCN Resnet 101, the accuracy is over the 90th percentile, while in the case of Faster RCNN with Inception V2, it falls below the 30th percentile. When the same method is employed to calculate the percentiles in terms of memory consumption, both Faster RCNN and R-FCN provide results within the 80th percentile. Although this justification is merely illustrative and it does not reference the complexity of DEA analysis, it shows the utility of this methodology to define efficiency frontiers in a multivariate data scenario of a high degree of complexity.

To conclude the analysis of the results for the managerial DEA model applied on the data obtained from the detection through the configurations of neural networks under study, we will compare the obtained results. The resolution of the original image, the total accuracy of the detection (mAP), the amount of memory used during the training and, finally, the number of needed operations to carry out the complete training process will be taken as reference. Based on the image resolution, the results presented in Table 3, and illustrated in Fig. 1, denote that methods based on R-FCN are much more stable than those based on Faster RCNN according to this input. This conclusion is obtained from the analysis of the average values of efficiency. Faster RCNN values for this output are relatively low (around 0.8) and unstable, unlike those of R-FCN, whose values are close to 1. These results lead to the conclusion that, regardless of the original image resolution, algorithms based on R-FCN obtain results that make them belong to the efficiency frontier. However, there are some configurations for Faster RCNN where efficiency coefficient is high. For example, Faster RCNN + MobileNet for an image resolution of 600 $\times$ 600, where the efficiency value is 1.

Same happens for the efficiency analysis in terms of accuracy, as shown in Fig. 2, and whose numeric values grouped by mAP intervals and network architecture are shown in Table 4. Nonetheless, in this case it can be observed that those methods with greater precision present an efficiency value close to 1, independently of whether it is R-FCN or Faster RCNN. For the next example, values where grouped by accuracy intervals for each architecture–extractor pair. Example of this are Faster RCNN MobileNet [19.3–22.3] and Faster RCNN Resnet 101 [22.3–25.3].

Concluding with the DEA managerial algorithm, Figs. 3 and 4, which show the existing correlation between the efficiency value and the consumed memory and the necessary operations, respectively, are presented. On one hand, In the first figure, the smaller the amount of memory consumed, the higher the efficiency values tend to be. On the other hand, in the second case, a greater number of operations leads to a decrease of these values. We should not forget that this is an analysis of grouped values, so the possibility that there are specific configurations that do not follow this identified patte rn is present.

Table 4
Summary of the efficiency coefficients of the DEA analysis (managerial/probabilistic/Bayesian) grouped by mAP intervals for each of the analysed architectures.

mAP accuracy interval	Average DEA coefficient man/prob/bay	Min DEA coefficient man/prob/bay	Max DEA coefficient man/prob/bay
Faster RCNN			
Inception Resnet V2			
[22.3–25.3]	0.814/0.845/0.780	0.628/0.691/0.577	1.000/1.000/0.983
[25.3–28.3]	0.756/0.844/0.719	0.663/0.781/0.595	0.861/0.932/0.910
[28.3–31.3]	0.692/0.748/0.637	0.584/0.652/0.354	0.907/0.947/0.934
[31.3–34.3]	0.805/0.901/0.822	0.675/0.794/0.641	0.923/0.972/0.965
[34.3–37.3]	0.843/0.958/0.713	0.705/0.913/0.404	0.993/1.000/0.978
Inception V2			
[13.3–16.3]	0.826/0.836/0.741	0.709/0.711/0.588	0.952/0.969/0.936
[16.3–19.3]	0.904/0.924/0.904	0.883/0.918/0.897	0.925/0.931/0.910
[19.3–22.3]	0.846/0.857/0.821	0.697/0.714/0.682	1.000/1.000/0.902
Inception V3			
[19.3–22.3]	0.946/0.999/0.978	0.892/0.997/0.960	1.000/1.000/0.997
[22.3–25.3]	0.760/0.799/0.715	0.623/0.654/0.450	0.936/1.000/0.960
[25.3–28.3]	0.959/1.000/0.994	0.959/1.000/0.994	0.959/1.000/0.994
[28.3–31.3]	0.843/0.876/0.800	0.628/0.629/0.528	1.000/1.000/0.967
MobileNet			
[13.3–16.3]	1.000/1.000/0.993	1.000/1.000/0.983	1.000/1.000/1.000
[16.3–19.3]	0.938/0.942/0.955	0.877/0.879/0.893	1.000/1.000/1.000
[19.3–22.3]	1.000/1.000/0.995	1.000/1.000/0.992	1.000/1.000/1.000
Resnet 101			
[22.3–25.3]	0.988/0.993/0.981	0.975/0.980/0.945	1.000/1.000/1.000
[25.3–28.3]	0.898/0.910/0.896	0.691/0.722/0.713	1.000/1.000/0.991
[28.3–31.3]	0.913/0.934/0.900	0.806/0.837/0.801	1.000/1.000/0.987
[31.3–34.3]	0.928/0.982/0.941	0.834/0.940/0.887	1.000/1.000/0.990
[34.3–37.3]	0.950/1.000/0.940	0.950/1.000/0.940	0.950/1.000/0.940
VGG			
[16.3–19.3]	1.000/1.000/0.970	1.000/1.000/0.970	1.000/1.000/0.970
[19.3–22.3]	1.000/1.000/0.997	1.000/1.000/0.994	1.000/1.000/1.000
[22.3–25.3]	0.997/1.000/0.991	0.993/1.000/0.973	1.000/1.000/1.000
[25.3–28.3]	0.968/1.000/0.983	0.937/1.000/0.972	1.000/1.000/0.994
R-FCN			
Inception Resnet V2			
[19.3–22.3]	1.000/1.000/0.997	1.000/1.000/0.997	1.000/1.000/0.997
[22.3–25.3]	0.988/0.989/0.950	0.931/0.937/0.806	1.000/1.000/1.000
[25.3–28.3]	0.980/0.988/0.912	0.934/0.941/0.728	1.000/1.000/0.974
[28.3–31.3]	0.979/0.980/0.924	0.897/0.898/0.643	1.000/1.000/0.998
[31.3–34.3]	1.000/1.000/0.8666	1.000/1.000/0.805	1.000/1.000/0.943
Inception V2			
[13.3–16.3]	0.996/0.999/0.981	0.977/0.995/0.890	1.000/1.000/1.000
[16.3–19.3]	1.000/1.000/0.957	1.000/1.000/0.957	1.000/1.000/0.957
[19.3–22.3]	0.992/1.000/0.979	0.975/1.000/0.961	1.000/1.000/0.998
MobileNet			
[13.3–16.3]	1.000/1.000/0.984	1.000/1.000/0.920	1.000/1.000/1.000
[16.3–19.3]	1.000/1.000/0.994	1.000/1.000/0.992	1.000/1.000/0.996
Resnet 101			
[19.3–22.3]	0.929/1.000/0.995	0.591/1.000/0.995	1.000/1.000/0.995
[22.3–25.3]	0.929/0.972/0.945	0.591/0.897/0.801	1.000/1.000/1.000
[25.3–28.3]	0.929/0.908/0.853	0.591/0.610/0.520	1.000/1.000/0.993
[28.3–31.3]	0.929/0.948/0.893	0.591/0.716/0.617	1.000/1.000/1.000
[31.3–34.3]	0.929/1.000/0.810	0.591/1.000/0.773	1.000/1.000/0.847

4.4. Probabilistic DEA analysis

Similar results to the previous ones are shown when applying the probabilistic DEA technique to the input data. In this case, the R-FCN instance belongs again to the efficiency frontier (DEA value equal to 1), while Faster RCNN maintains a similar value, although lightly higher, in the result of the analysis methodology. Another difference between the results of the probabilistic DEA and the Managerial applied previously is the number of DMUs that belong to the efficiency frontier. This number of DMUs has increased from 69 to 85 (18%). This is due to the inclusion of a tolerance margin for the probabilistic DEA model, since this model assumes that data follow a probabilistic distribution.

Figs. 5–8 illustrate the probabilistic DEA analysis results. As these figures show, correlation between both algorithms (probabilistic and managerial) exists. However, there is a clear difference between both models. According to the probabilistic DEA model, efficiency values are subtly higher than those obtained with the managerial method. Except this, the previous conclusions and the existing correlation between the inputs and outputs of the model are also maintained in the case of probabilistic DEA.

These results can also be found in Table 4, where, as in the previous section, an analysis of the efficiency values obtained by performing the probabilistic DEA analysis grouped by average accuracy of the architectures.

4.5. Bayesian DEA analysis

Finally, the Bayesian DEA model is applied to the same data set. This time, the same two previous DMUs (Faster RCNN architecture with Inception V2 with an image resolution of 600 pixels and 300 proposals sent to box classifier and R-FCN architecture with Resnet 101 with an image resolution of 300 pixels and 20 proposals sent to box classifier) have been compared one more time. This execution shows that both efficiency values have decreased and none of the two units belongs to the efficiency frontier. However, the R-FCN unit shows a very efficient value (0.9994). Comparing the whole set of efficient DMUs, the results present the same trend as in the direct comparison between the two selected DMUs: the number of efficient values decreased from 85 to only 6. As discussed above during the formal description of the Bayesian DEA analysis method, this is the result of an iterative execution of the DEA with a lightly variation in the data set (both inputs and outputs) that follow the probabilistic distribution also described in this section. This iterative execution increases the probability that (1) the results of DEA analysis in a specific DMU could obtain values different than 1 in, at least one iteration; and (2) when modifying the inputs and outputs, the resulting value of the analysis will vary slightly, enough for the exclusion of certain DMUs from the efficiency frontier.

Finally, Figs. 9–12 represent the results of the previous process using the Bayesian DEA model. Unlike the probabilistic model, it can be observed how the number of efficient units has been significantly reduced this time. In any case, the figures show similar patterns to those obtained previously, denoting that architectures based on R-FCN have higher values for efficiency, many of them belonging to the efficiency frontier. In contrast, by reducing the number of DMUs in the efficiency frontier, the comparison capacity between the architectures is improved, enabling the obtention of those configurations that maximise efficiency in a more robust and stable way than with the previous techniques based on probabilistic and managerial DEA.

These results can also be found in Table 4, where the analysis of the efficiency values obtained by the Bayesian DEA according to the average accuracy of the architectures has been performed.

4.6. Efficiency projections

DEA not only determines the level of efficiency of the current DMU based on the rest of the DMUs analysed, but also makes it possible to identify which parameters of the proposed architecture should be modified to make the system more efficient: first by modifying input values, and then the outputs if necessary.

Table 5 shows the correction values for DMU #35, which had an initial efficiency of 61%. In this table can be seen how DEA provides with a set of corrections on the original values to convert this DMU into an efficient unit. The modified variable appears in the "Variable" column. The columns "original value" and "radial movement"

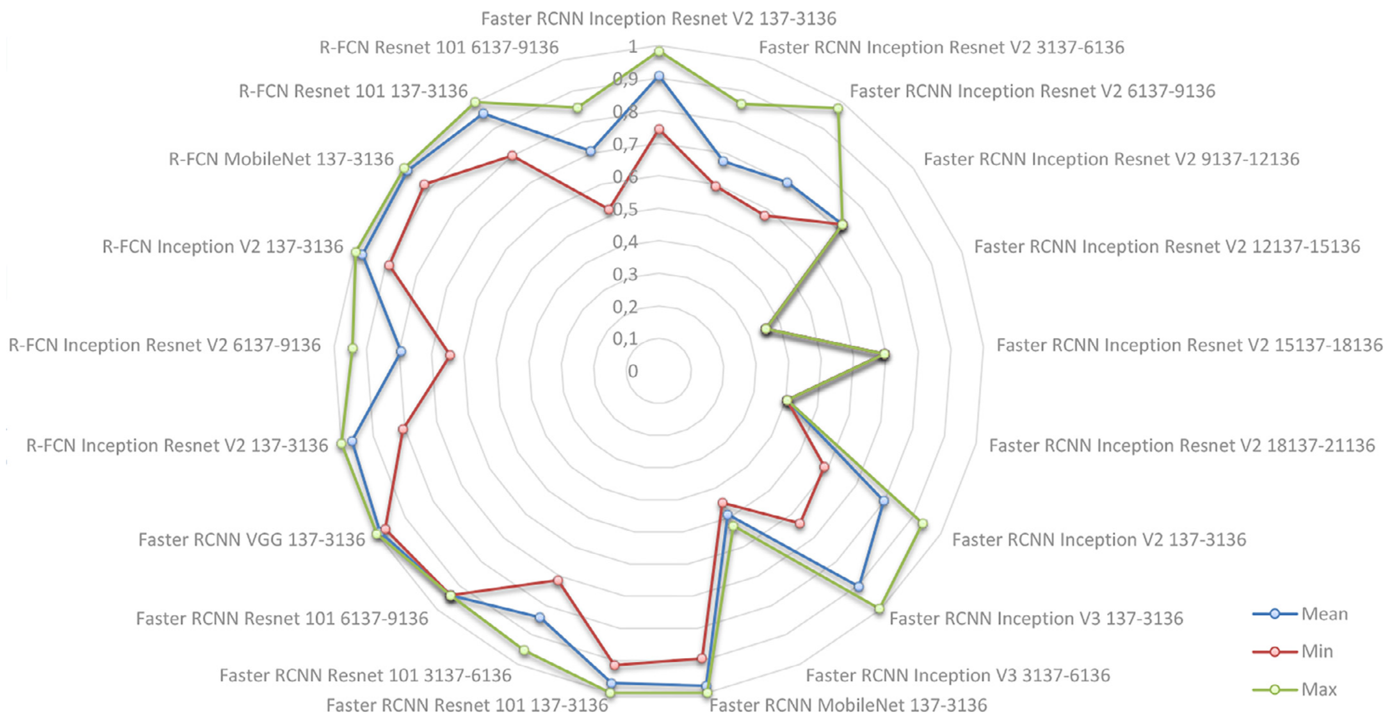

Fig. 7. DEA probabilistic method maximum, minimum and average efficiency over different detection techniques based on the consumed memory.

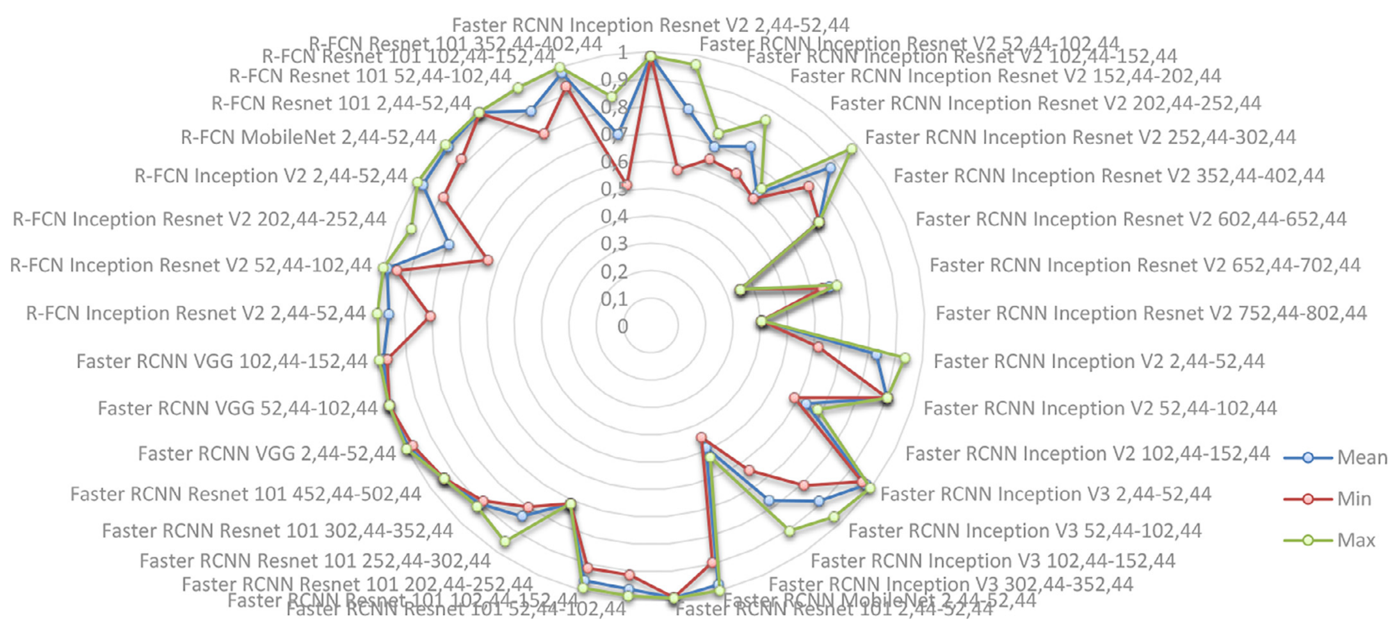

Fig. 8. DEA probabilistic method maximum, minimum and average efficiency over different detection techniques based on the number of operations required to complete the whole process.

contain the initial value of the variable and the percentage of correction on such value, respectively. The "slack movement" column reflects the absolute value of the resulting correction and, finally, "projected value" column presents the value that this variable should have in that unit to become efficient. Some of these values may not be consistent, as for example a negative value for used memory, as is the case shown in Table 5. This is because the DEA model employed does not offer restrictions on the input and output values, which could result in inconsistent values. This fact must be interpreted at a high level, getting to reduce that value as much as possible whenever possible. If these values are tuned and they approach the optimal value of the projection, the efficiency will increase.

To continue, the correlation of the previously selected DMU with the rest of the DMUs will be compared. By means of this process, the behaviour of inputs and outputs of other DMUs can be "mocked" in order to get the initial DMU to turn efficient, performing a mix between the different behaviours of efficient DMUs. Table 6 shows the results of the equivalence analysis. In this analysis, the DMUs related to a given unit are considered,

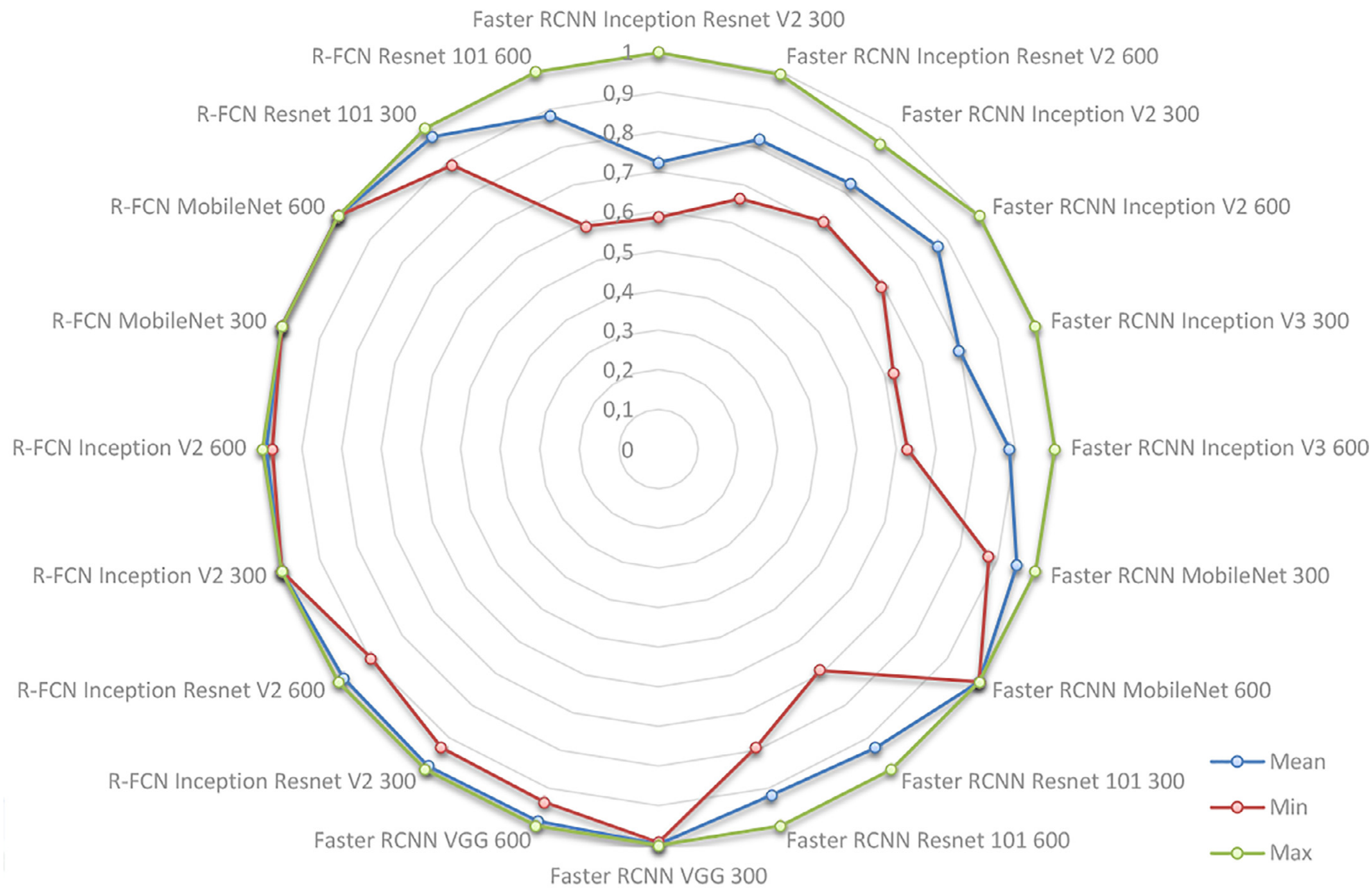

Fig. 9. DEA Bayesian method maximum, minimum and average efficiency over different detection techniques based on the resolution of the input images.

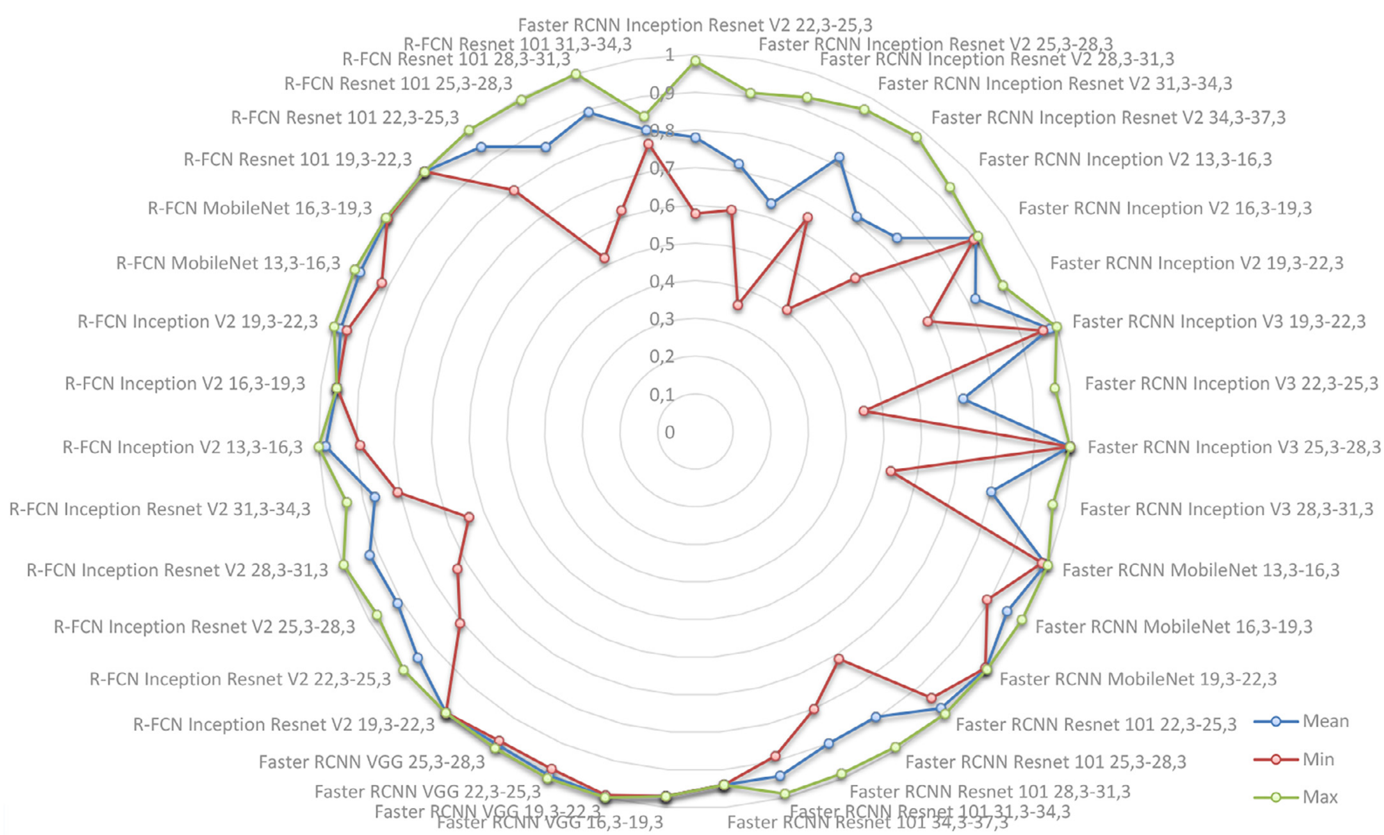

Fig. 10. DEA Bayesian method maximum, minimum and average efficiency over different detection techniques based on the box detection (mAP) accuracy.

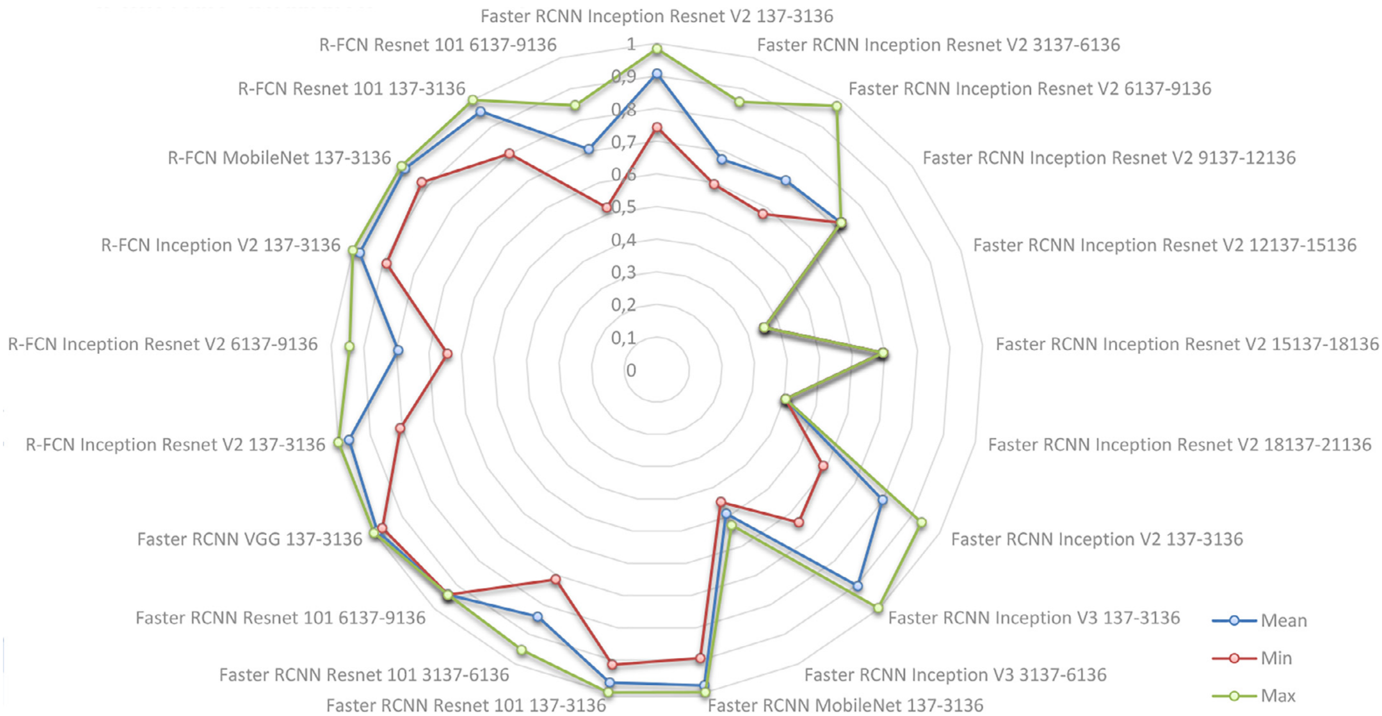

Fig. 11. DEA Bayesian method maximum, minimum and average efficiency over different detection techniques based on the consumed memory.

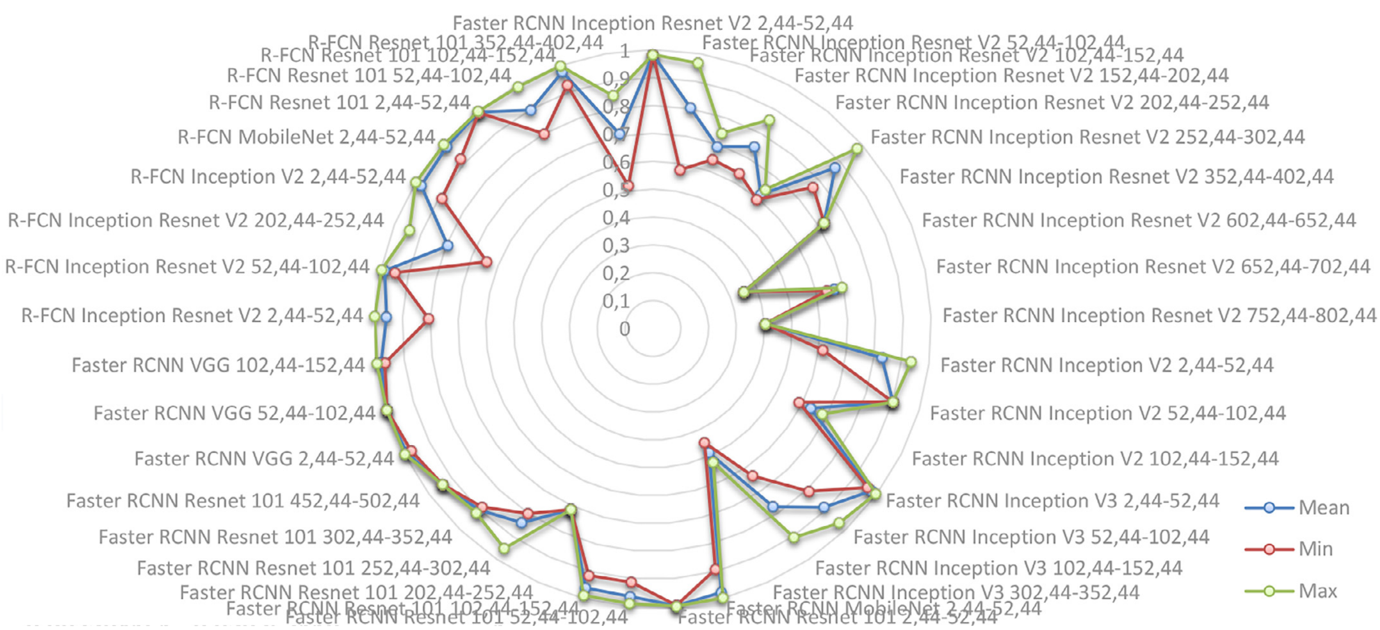

Fig. 12. DEA Bayesian method maximum, minimum and average efficiency over different detection techniques based on the number of operations required to complete the whole process.

so that the percentage of simulation that the selected unit must have to the reference unit is established so that the first one is considered efficient. This allows to establish criteria of change on the data, so that if we want a DMU to become efficient, it is possible to establish how much that DMU should be similar to the rest of the units to achieve that efficiency. The DEA analysis does not directly indicate which are the concrete attributes that must be transformed when establishing that direct equivalence with another DMU, but it is easy to establish this criterion given the optimisation data previously seen.

Finally, the whole set of architectures will be processed in order to identify which inputs and outputs should be optimised (increased in case of positive inputs or outputs and decreased otherwise) in order to transform the DMUs into efficient units through the aforementioned optimisation process. While it is true that the outputs can not be directly changed, this analysis allows

Table 5
Corrections proposed for DMU #35 (Faster RCNN inception V2).

Results for DMU #35
Managerial efficiency = 0.6162
Projection summary:

Variable		Original value	Radial movement	Slack movement	Projected value
Input	Resolution	300	−8%	−24.82	275.18
Input	# Proposals	100	−181%	−181.28	−81.28
Output good	Overall mAP	15.3	+21%	+3.19	18.49
Output good	mAP large	27.3	+1%	0.30	27.60
Output good	mAP medium	7.4	+0.5%	0.04	7.44
Output good	mAP small	0.5	+2.5%	0.01	0.51
Output good	mAP @. 75IOU	14.7	+0.8%	0.12	14.82
Output good	mAP @. 50IOU	18.5	+0.8%	0.22	28.72
Output bad	# parameters	13.31	−2.9%	−0.39	13.70
Output bad	Memory	861	−100%	−861.17	−0.17
Output bad	Memory std	51	−13%	−6.63	44.37
Output bad	CPU	535	−81.6%	−436.62	98.38
Output bad	CPU std	7	−5.9%	−0.42	7.42
Output bad	GPU	118	−21.8%	−25.67	143.67
Output bad	GPU std	15	−0.5%	−0.08	15.08
Output bad	Flops	41.95	−15.3%	−6.40	35.55

Listing of peers:

Peer	Lambda weight (%)
#72	13
#73	9
#131	11
#132	15
#133	10

Table 6
Equivalence percentage between 16 selected DMUs (columns) and the rest of the DMUs considered (rows) to transform the selected DMU into an efficient unit.

	#11 (%)	#31 (%)	#34 (%)	#35 (%)	#36 (%)	#41 (%)	#45 (%)	#46 (%)	#51 (%)	#61 (%)	#62 (%)	#65 (%)	#66 (%)	#99 (%)	#100 (%)
#9	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
#12	1	0	2	1	0	4	8	1	6	1	17	9	0	0	2
#16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
#19	1	0	0	0	0	0	0	4	0	6	0	2	3	21	15
#23	2	0	0	0	0	3	2	1	2	2	1	6	0	1	2
#24	2	0	0	0	2	3	1	1	4	2	1	10	1	1	2
#26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
#31	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0
#50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
#56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
#72	1	4	17	13	0	3	6	0	2	1	5	2	0	0	1
#73	1	3	12	9	0	3	5	0	2	0	4	1	0	0	1
#78	2	3	3	5	6	4	5	6	9	3	9	11	3	2	6
#81	44	6	1	2	31	35	10	30	31	40	11	11	4	0	2
#82	8	1	1	1	2	7	14	6	4	10	9	9	1	2	6
#83	5	0	1	1	1	5	14	3	2	6	9	7	1	1	6
#84	4	1	1	1	0	3	12	1	0	4	4	2	0	1	4
#86	4	0	0	0	6	0	0	18	2	10	0	1	38	7	2
#87	1	0	0	0	3	0	0	22	11	6	1	12	72	28	15
#88	2	0	0	0	1	0	0	5	0	9	0	2	9	43	28
#91	4	0	0	0	7	3	0	5	15	5	0	2	8	0	0
#101	6	24	3	7	4	12	3	2	6	5	1	1	1	0	0
#106	4	4	0	1	11	5	1	8	12	4	1	1	2	0	0
#116	5	0	0	0	3	1	0	4	0	7	1	1	4	0	0
#117	2	0	0	0	2	1	0	8	9	3	4	10	19	4	3
#119	0	0	0	0	0	0	0	2	12	1	4	9	7	2	2
#131	8	36	5	11	8	17	4	3	11	5	1	1	1	0	0
#132	2	8	13	15	1	6	8	1	4	2	3	2	0	0	1
#133	0	3	11	10	0	3	4	0	2	1	2	1	0	0	0

the establishment of an objective criterion which improves the efficiency of the considered architectures.

5. Conclusions and future work

In this paper we propose a low-computing-demanding model based on DEA for the measurement of the efficiency (performance) of different object-detection architectures, and to provide a set of tools for the improvement of different configurations of neural networks based on the most relevant input and output parameters. Specifically, 139 configurations of neural networks used for object detection in images have been processed based on 16 parameters: 3 inputs and 13 outputs. Three different techniques for the computation of efficiency have been presented for this analysis in order to increase the robustness: Managerial DEA, Probabilistic DEA and Bayesian DEA. The three techniques have been applied on

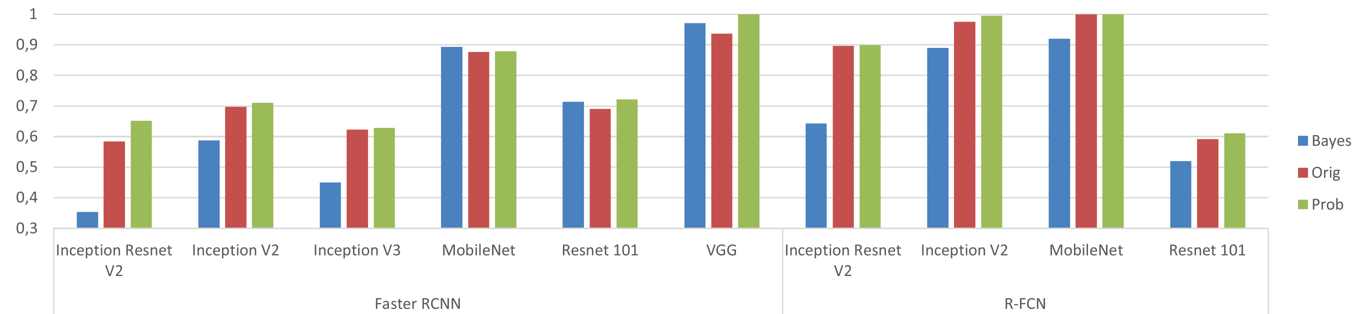

Fig. 13. Average efficiency values for each of the detection techniques grouped by the type of applied DEA algorithm (Managerial, probabilistic or Bayesian).

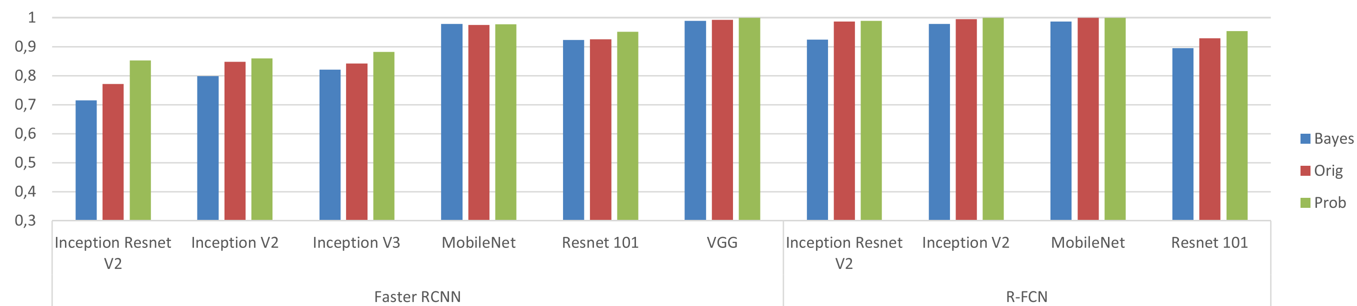

Fig. 14. Minimum efficiency values for each of the detection techniques grouped by the type of applied DEA algorithm (Managerial, probabilistic or Bayesian).

several neural-network configurations based on R-FCN Inception V2, R-FCN MobileNet, R-FCN Resnet and R-FCN Inception Resnet which places them at the optimization frontier for inputs and outputs values presented. It is also demonstrated after the application of these efficiency calculation and projection techniques that, as could be expected, R-FCN Inception Resnet generally obtains the highest values in terms of efficiency. Regarding the different DEA techniques introduced in this paper, it should be noticed that the probabilistic DEA, in general, tends to maximise the number of DMUs present in the efficiency frontier, unlike the Bayesian DEA, which reduces the number of units in such frontier. The latter allows, therefore, the selection of those DMUs that belong to the efficiency set in a more restrictive way, and, in consequence, to point out the optimum efficiency of the rest of the units. This can be seen in Figs. 13 and 14. As mentioned in Section 3, one of the original DEA limitations is that since it is an exclusively mathematical technique, random noise effects over data source are not considered in this DEA approach. In the probabilistic and Bayesian DEA, a statistical approach is adopted and therefore these factors are taken into account, when considering the observed data as collection of random variables. As we can see, the results obtained when applying the three techniques are similar, which is an indication of robustness, that is, the results are not sensitive to random variation effects. On the other hand, in the results obtained after the application of the algorithms, we verify that the standard errors of the Bayesian estimated efficiencies, in general, are small enough compared to the estimated values (that is, the estimations are very accurate), which means also that the results obtained are robust against possible random effects. Finally, this novel comparison methodology for units based on neural-network architectures, allows to determine whether a given architecture belongs to the most-efficient architectures set or may be improved based on the values of its inputs and outputs. In this way, this methodology also enables the determination of the input values to be improved or the values of the outputs that must be reached in order to make efficient a non-efficient unit. It should be noted that the calculation of the efficiency frontier using the presented DEA techniques is computationally undemanding, so it is possible to feed back the data set and launch the process of computing the frontier without major inconveniences. In contrast, the computational and temporal cost of training a neural network based on the trial-error method is extremely high, hence the usefulness of this exposed methodology. New meta-architectures [47–49] may employ this method to determine their relative efficiency and check which input can be improved to achieve the frontier results. In this way, a specific architecture could be chosen based on the characteristics of the proposed inputs, giving a fairly clear approximation of how the selected architecture will behave based on the features of the given inputs.

As future work, this paper sets a milestone for a new research track which may be expanded in the following three main directions:

- Application of DEA models to more recent object detection architectures so that the results and conclusions of this work can be confirmed or contrasted;
- Development of models which may employ DEA models to improve the efficiency for the training of neural networks; and
- Development of expert systems equipped with low-computing-demanding DEA models for the dynamic application of object detection architectures depending on the scenario.

Intellectual property

We confirm that we have given due consideration to the protection of intellectual property associated with this work and that there are no impediments to publication, including the timing of publication, with respect to intellectual property. In so doing we confirm that we have followed the regulations of our institutions concerning intellectual property.

Research ethics

We further confirm that any aspect of the work covered in this manuscript that has involved human patients has been conducted with the ethical approval of all relevant bodies and that such approvals are acknowledged within the manuscript.

Authorship

We confirm that the order of authors listed in the manuscript has been approved by all named authors. We confirm that the order of authors listed in the manuscript has been approved by all named authors.

Declaration of Competing Interest

None.

Acknowledgment

This research is funded by the Spanish Ministry of Economy and Competitiveness through the projects Vision and Crowdsensing Technology for an Optimal Response in Physical-Security (TIN2017-82113-C2-1-R)

Appendix A. Data envelopment analysis formal description

A.1. Natural and managerial disposability

In model (A.1), each jth DMU $j = 1, \ldots, n$, uses inputs $X_j = (x_{1j}, \ldots, x_{mj})^T$ to produce desirable outputs $G_j = (g_{1j}, \ldots, g_{sj})^T$ and undesirable outputs $B_j = (b_{1j}, \ldots, b_{hj})^T$. Furthermore, d_i^x, $i = 1, \ldots, m$, d_r^g, $r = 1, \ldots, s$ and d_f^b, $f = 1, \ldots, h$ are all slack variables related to inputs, desirable outputs and undesirable outputs, respectively. $\lambda = (\lambda_1, \ldots, \lambda_n)^T$ are structural or intensity variables, which are unknown and are used for connecting the input and output vectors by a convex combination. R is the range resolute through the upper and lower bounds of inputs, desirable outputs and undesirable outputs, expressed by:

$$R_i^x = (m+s+h)^{-1} \left(\max\{x_{ij}/j = 1, \ldots, n\} - \min\{x_{ij}/j = 1, \ldots, n\}\right)$$

$$R_r^g = (m+s+h)^{-1} \left(\max\{g_{rj}/j = 1, \ldots, n\} - \min\{g_{rj}/j = 1, \ldots, n\}\right)$$

and

$$R_f^b = (m+s+h)^{-1} \left(\max\{b_{fj}/j = 1, \ldots, n\} - \min\{b_{fj}/j = 1, \ldots, n\}\right)$$

The natural efficiency of the kth policy is evaluated by the following CRS and VRS radial model (see [50] for a better understanding):

$$\begin{aligned}
\max\ & \xi + \epsilon \left(\sum_{i=1}^{m} R_i^x d_i^x + \sum_{r=1}^{s} R_r^g d_i^g + \sum_{f=1}^{h} R_f^b d_i^b \right) \\
s.t.\ \sum_{j=1}^{n} x_{ij}\lambda_j + d_i^x &= x_{ik}\,, i = 1, \ldots, m\,, \\
\sum_{j=1}^{n} g_{rj}\lambda_j - d_r^g - \xi g_{rk} &= g_{rk}\,, r = 1, \ldots, s\,, \\
\sum_{j=1}^{n} b_{fj}\lambda_j + d_f^b + \xi b_{fk} &= b_{fk}\,, f = 1, \ldots, h\,, \\
d_i^x &\geq 0\,, i = 1, \ldots, m\,, \\
d_r^g &\geq 0\,, r = 1, \ldots, s\,, \\
d_f^b &\geq 0\,, f = 1, \ldots, h\,, \\
\xi &\quad \text{Unrestricted}
\end{aligned} \tag{A.1}$$

where: ξ, which is an unrestricted parameter, represents an unknown inefficiency score indicating a distance between the efficiency frontier and a particular observed vector of desirable and undesirable outputs; In our case, ϵ takes a value of 0.0001 to reduce the influence of slack variables for computation reasons. If the restriction $\sum_{j=1}^{n} \lambda_j = 1$ is added to the model (A.1), then the VRS model is obtained (Model A.1*).

The first restriction in systems (A.1) and (A.1*) seeks values of λ_j to construct a composite unit, with inputs such that $\sum_{j=1}^{n} x_{ij}\lambda_j = -d_i^x + x_{ik}\,, i = 1, \ldots, m$. The positive slack variables d_i^x indicate that further decreases in inputs may be performed, which necessarily alter the proportions used, and hence they show the existence of inefficiencies.

Similarly, the second restriction, $\sum_{j=1}^{n} g_{rj}\lambda_j = d_r^g + \xi g_{rk} + g_{rk}\,, r = 1, \ldots, s$, tells us that the desirable outputs may be increased or at least maintained by performing a radial expansion ξg_{rk} and an increase in the slack variable d_r^g.

Analogously, the third restriction, $\sum_{j=1}^{n} b_{fj}\lambda_j = -d_f^b - \xi b_{fk} + b_{fk}\,, f = 1, \ldots, h$, indicates that we have decreased the inputs, and then we reduce the undesirable outputs both radially and in their slack variables.

The objective function implies that two sources of inefficiency may be identified. Ak-policy is efficient if and only if the following two conditions are satisfied: (a) $\xi = 0$; (b) $d_i^x = 0, d_r^g = 0, d_f^b = 0$. In this efficient case, k-policy belongs to the efficiency frontier. It fulfils restrictions in equation systems (A.1) and (A.1*), and therefore the objective function is 0. Otherwise, if it is not efficient, the value of the objective function is greater than 0, due to possible radial movement and possible displacements in the slack variables.

The natural efficiency is then measured as follows:

$$\theta^* = 1 - \left[\xi^* + \epsilon \left(\sum_{i=1}^{m} R_i^x d_i^{x^*} + \sum_{r=1}^{s} R_r^g d_i^{g^*} + \sum_{f=1}^{h} R_f^b d_i^{b^*} \right) \right]$$

This metric of unified efficiency score takes values between 0 and 1. If the k-policy is efficient, then the objective function of equation systems (A.1) and (A.1*) is 0, and hence the efficiency score equals $\theta^* = 1$. All slack variables obtained in the optimality of models (A.1) and (A.1*) indicate the level of inefficiency [50].

The Managerial efficiency of the kth policy is evaluated by the following CRS and VRS radial models [50]:

$$\begin{aligned}
s.t.\ \sum_{j=1}^{n} x_{ij}\lambda_j - d_i^x &= x_{ik}\,, i = 1, \ldots, m\,, \\
\sum_{j=1}^{n} g_{rj}\lambda_j - d_r^g - \xi g_{rk} &= g_{rk}\,, r = 1, \ldots, s\,, \\
\sum_{j=1}^{n} b_{fj}\lambda_j + d_f^b + \xi b_{fk} &= b_{fk}\,, f = 1, \ldots, h\,, \\
d_i^x &\geq 0\,, i = 1, \ldots, m\,, \\
d_r^g &\geq 0\,, r = 1, \ldots, s\,, \\
d_f^b &\geq 0\,, f = 1, \ldots, h\,, \\
\xi &\quad \text{Unrestricted}
\end{aligned} \tag{A.2}$$

Similarly, if the restriction $\sum_{j=1}^{n} \lambda_j = 1$ is added to Model (A.2), then we obtain a VRS Model (A.2*). VRS models can provide with the returns to scale (RTS) and damage to scale (DTS) metrics (see [50] for a better understanding). It is clear that for the natural efficiency the returns to scale has to be increased and for managerial efficiency the damages to scale has to be decreased. Otherwise the technical units are not working well and should correct the imbalances, using the information of the efficient units to which they have to be similar (peers).

A.2. Chance-constrained DEA.

In the formulation of chance-constrained DEA we consider the more general case in which both inputs and outputs are random variables. Then the problem associated to Natural DEA becomes:

$$
\begin{aligned}
&\max\ \xi\\
&s.t.\\
&P\left(\sum_{j=1}^{n} x_{ij}\lambda_j \le x_{ik}\right) \ge 1-\alpha, \quad i=1,\ldots,m\\
&P\left(\sum_{j=1}^{n} g_{rj}\lambda_j \ge (1+\xi)g_{rk}\right) \ge 1-\alpha, \quad r=1,\ldots,s\\
&P\left(\sum_{j=1}^{n} b_{fj}\lambda_j \le (1-\xi)b_{fk}\right) \ge 1-\alpha, \quad f=1,\ldots,h\\
&\xi \text{ Unrestricted}
\end{aligned} \tag{A.3}
$$

where $1-\alpha$ denotes the minimum probability with which the constraints are satisfied. The most common choice is $\alpha = 0.05$ (other common values are $\alpha = 0.01$ or $\alpha = 0.1$). It is also possible to take different α values for each constraint.

Some inputs may be deterministic in some particular environments. In such a case, we just have to remove the probability term in the corresponding restrictions. In fact, many applications of chance-constrained DEA consider that the inputs are deterministic and only the outputs are stochastic.

We assume the usual hypothesis that both inputs and outputs follows a multivariate normal distribution, and we use the notation μ_{ij}^x and σ_{ij}^x for the expected value and the standard deviation of x_{ij}, that is, $x_{ij} \sim N(\mu_{ij}^x, (\sigma_{ij}^x)^2)$. Then, the restrictions of problem (A.3) are converted to certainty equivalents [45,46,51], as follows.

Letś consider the input restrictions $P(\sum_{j=1}^{n} x_{ij}\lambda_j \le x_{ik}) \ge 1-\alpha, \quad i=1,\ldots,m$. Using the notation $s_{ik} = \sum_{j=1}^{n} x_{ij}\lambda_j - x_{ik}$ and $\phi(\bullet)$ for the cumulative distribution function of the standard normal distribution, the ith restriction may be denoted as:

$$
\begin{aligned}
P(s_{ik} \le 0)\ \ge\ 1-\alpha \Leftrightarrow P\left(\frac{s_{ik}-E[s_{ik}]}{\sqrt{\mathrm{var}[s_{ik}]}} \le \frac{-E[s_{ik}]}{\sqrt{\mathrm{var}[s_{ik}]}}\right) \ge 1-\alpha\\
\Leftrightarrow \phi\left(\frac{-E[s_{ik}]}{\sqrt{\mathrm{var}[s_{ik}]}}\right) \ge 1-\alpha \Leftrightarrow \frac{-E[s_{ik}]}{\sqrt{\mathrm{var}[s_{ik}]}} \ge \phi^{-1}(1-\alpha)\\
\Leftrightarrow E[s_{ik}] + \phi^{-1}(1-\alpha)\sqrt{\mathrm{var}[s_{ik}]} \le 0
\end{aligned}
$$

Because of $E[s_{ik}] = \sum_{j=1}^{n} E\left[x_{ij}\right]\lambda_j - E[x_{ik}]$, the input restrictions become:

$\sum_{j=1}^{n} E\left[x_{ij}\right]\lambda_j + \phi^{-1}(1-\alpha)\sqrt{\mathrm{var}[s_{ik}]} \le E[x_{ik}], \quad i=1,\ldots,m.$

Under the general multivariate normal assumption, a non-linear deterministic equivalent is obtained due to the term $\sqrt{\mathrm{var}[s_{ik}]}$. The computing time necessary to solve the problem increases considerably. Moreover, simulation techniques are needed for the application of the Bayesian methodology and the problem has to be solved thousands of times. Therefore, the computing time is crucial in practice. We assume the single factor disturbance model of [51] to face this challenge. More explicitly:

$x_{ij} = \mu_{ij}^x + \sigma_{ij}^x\eta^x, \quad i=1,\ldots,m, \quad j=1,\ldots,n$ where η^x is the only factor related to the inputs, and follows a standard normal distribution. From this hypothesis it is clear that the mean and standard deviation of x_{ij} are μ_{ij}^x and σ_{ij}^x, respectively. Under the single factor disturbance hypothesis, the deterministic equivalent is expressed in terms on absolute value. Specifically:

$$
\begin{aligned}
\mathrm{var}[s_{ik}]\ =\ \mathrm{var}\left[\left(-\sigma_{ik}^x + \sum_{j=1}^{n}\sigma_{ij}^x\lambda_j\right)\eta^x\right] = \left(-\sigma_{ik}^x + \sum_{j=1}^{n}\sigma_{ij}^x\lambda_j\right)^2\\
\Rightarrow \mathrm{var}\sqrt{[s_{ik}]} = \left|-\sigma_{ik}^x + \sum_{j=1}^{n}\sigma_{ij}^x\lambda_j\right|
\end{aligned}
$$

Therefore, the input constraints may be expressed as:

$$
\sum_{j=1}^{n}\mu_{ij}^x\lambda_j + \phi^{-1}(1-\alpha)\left|-\sigma_{ik}^x + \sum_{j=1}^{n}\sigma_{ij}^x\lambda_j\right| \le \mu_{ik}^x, \quad i=1,\ldots,m
$$

This problem can be transformed into an ordinal linear programming problem [51,52]. First, we may consider:

$$
q_{1i}^X = \begin{cases} -\sigma_{ik}^x + \sum_{j=1}^{n}\sigma_{ij}^x\lambda_j & if\ \ -\sigma_{ik}^x + \sum_{j=1}^{n}\sigma_{ij}^x\lambda_j \ge 0\\ 0 & if\ \ -\sigma_{ik}^x + \sum_{j=1}^{n}\sigma_{ij}^x\lambda_j < 0 \end{cases}, \quad i=1,\ldots,m
$$

$$
q_{2i}^X = \begin{cases} 0 & if\ \ -\sigma_{ik}^x + \sum_{j=1}^{n}\sigma_{ij}^x\lambda_j \ge 0\\ \sigma_{ik}^x - \sum_{j=1}^{n}\sigma_{ij}^x\lambda_j & if\ \ -\sigma_{ik}^x + \sum_{j=1}^{n}\sigma_{ij}^x\lambda_j < 0 \end{cases}, \quad i=1,\ldots,m
$$

Hence, $|-\sigma_{ik}^x + \sum_{j=1}^{n}\sigma_{ij}^x\lambda_i|$ can be expressed by $q_{1i}^x + q_{2i}^x$, where q_{1i}^x and q_{2i}^x satisfy $q_{1i}^x - q_{2i}^x = -\sigma_{ik}^x + \sum_{j=1}^{n}\sigma_{ij}^x\lambda_i$ and $q_{1i}^x q_{2i}^x = 0$; moreover, $q_{1i}^x \ge 0, q_{2i}^x \ge 0$.. Thus, the input constraints have to be replaced by:

$$
\left.\begin{aligned}
&\sum_{j=1}^{n}\mu_{ij}^x\lambda_j + \phi^{-1}(1-\alpha)\left(q_{1i}^x + q_{2i}^x\right) \le \mu_{ik}^x\\
&-\sigma_{ik}^x + \sum_{j=1}^{n}\sigma_{ij}^x\lambda_j = q_{1i}^x - q_{2i}^x\\
&q_{1i}^x \ge 0, \quad q_{2i}^x \ge 0
\end{aligned}\right\} i=1,\ldots,m
$$

and then we may include $-\varepsilon\left(q_{1i}^x + q_{2i}^x\right)$ to the objective function to guarantee $q_{1i}^x q_{2i}^x = 0$ without modifying the optimal solution (where ε is the "non-Archimedean" infinitesimal quantity).

The application of the same procedure to good and bad outputs (for good outputs there is a minor difference because the constraint is of type greater than or equal, but the procedure is very similar), the deterministic linear equivalent problem to be solved is:

$$
Max\ \ \xi - \varepsilon\left(\sum_{i=1}^{m}\left(q_{1i}^x + q_{2i}^x\right) + \sum_{r=1}^{s}\left(q_{1r}^g + q_{2r}^g\right) + \sum_{f=1}^{h}\left(q_{1f}^b + q_{2f}^b\right)\right)
$$

$s.t.$

$$
\left.\begin{aligned}
&\sum_{j=1}^{n}\mu_{ij}^x\lambda_j + \phi^{-1}(1-\alpha)\left(q_{1i}^x + q_{2i}^x\right) \le \mu_{ik}^x\\
&-\sigma_{ik}^x + \sum_{j=1}^{n}\sigma_{ij}^x\lambda_j = q_{1i}^x - q_{2i}^x\\
&q_{1i}^x \ge 0, \quad q_{2i}^x \ge 0
\end{aligned}\right\} i=1,\ldots,m
$$

$$
\left.\begin{aligned}
&\sum_{j=1}^{n}\mu_{rj}^g\lambda_j - \phi^{-1}(1-\alpha)\left(q_{1r}^g + q_{2r}^g\right) \ge (1+\xi)\mu_{rk}^g\\
&-(1+\xi)\sigma_{rk}^g + \sum_{j=1}^{n}\sigma_{rj}^g\lambda_j = q_{1r}^g - q_{2r}^g\\
&q_{1r}^g \ge 0, \quad q_{2r}^g \ge 0
\end{aligned}\right\} r=1,\ldots,s
$$

$$
\left.\begin{aligned}
&\sum_{j=1}^{n}\mu_{fj}^b\lambda_j + \phi^{-1}(1-\alpha)\left(q_{1f}^b + q_{2f}^b\right) \le (1-\xi)\mu_{fk}^b\\
&-(1-\xi)\sigma_{fk}^b + \sum_{j=1}^{n}\sigma_{fj}^b\lambda_j = q_{1f}^b - q_{2f}^b\\
&q_{1f}^b \ge 0, \quad q_{2f}^b \ge 0
\end{aligned}\right\} f=1,\ldots,h
$$

ξ Unrestricted

The deterministic equivalent problem depends on μ_{ij}^x, σ_{ij}^x, μ_{rj}^g, σ_{rj}^g, μ_{fj}^b, σ_{fj}^b. We may estimate these quantities in the unusual case in which we are able to replicate the data in similar conditions, and supposing that the frontier remains constant for the different replications. Notwithstanding, we usually have only one replication and then we have to select values for the unknown parameters. The natural hypothesis for means is $\mu^x_{ij} = x_{ij},\ \mu^g_{rj} = g_{rj},\ \mu^b_{fj} = b_{fj}$. The standard deviations must be fixed by experts taking into account the possible magnitude of random variations in the data under consideration [46].

For Managerial DEA, the procedure is similar. Only the input restrictions are modified and become:

$$\left.\begin{array}{l} \sum\limits_{j=1}^{n} \mu^x_{ij}\lambda_j - \phi^{-1}(1-\alpha)\left(q^x_{1i} + q^x_{2i}\right) \geq \mu^x_{ik} \\ -\sigma^x_{ik} + \sum\limits_{j=1}^{n} \sigma^x_{ij}\lambda_j = q^x_{1i} - q^x_{2i} \\ q^x_{1i} \geq 0, \quad q^x_{2i} \geq 0 \end{array}\right\} \quad i = 1, \ldots, m$$

A.3. Bayesian DEA.

In Bayesian inference, the uncertainty related to unknown parameters in a statistical model is handled by the assumption of an existing distribution for it. By combining this prior function with the likelihood function via the Bayes theorem, we obtain the posterior distribution of the unknown parameters (conditional on the observed data), and this posterior distribution allows the obtention of inferences about both the unknown parameters and any function of them.

Usually, the posterior distribution is highly complex and analytical evaluation is not available. Therefore, the application of simulation-based approaches, such as the Gibbs sampling algorithm, [53–55] becomes necessary. It is important to remark that the Bayesian approach does not require the utilisation of the hypothetical large-sample constructs used in sampling-theory treatments, such as Bootstrap approach. The results of Bayesian inference are exact for finite samples, apart from simulation error.

We consider the inputs (for outputs the described process is similar) in order to specify the prior function and to obtain the posterior distribution. For simplicity in the notation, we will eliminate the upper script "*x*". As aforementioned, our deterministic-equivalent problem depends on μ_{ij} and σ_{ij}, but not on x_{ij}. Hence, the assumption of the hypothesis of inputs having a common mean is very restrictive, because in this case all the DMUs would have the same values of inputs in the deterministic equivalent problem. Consequently, we will consider $E[x_{ij}] = \mu_{ij}$.

If replicated data is available, $x_{ij} \sim N(\mu_{ij}, \sigma^2_{ij})$ may be assumed, which enables us to apply the Bayesian framework for Normal distribution using a non-informative prior distribution [56]. In the usual case (not replicated data), the introduction of prior information about the standard deviations is required, since there is not such information available in the data.

First, we may assume a common variance for each input, that is, $x_{ij} \sim N\left(\mu_{ij}, \sigma^2_i\right)$, and second, we suppose a uniform distribution for the variance. The necessary information for σ^2_i is not a single value, but a range of values representing variations of it. Specifically, the prior distribution is $\sigma^2_i \sim U\left(\sigma^2_{i0}, \sigma^2_{i1}\right)$, where $U(a, b)$ represents a uniform distribution in the interval (a, b). σ_{i0} and σ_{i1} are to be fixed by experts which take into account the possible magnitude of random variations in the data. For the means, we assume the usual non-informative Jeffreys prior distribution $\pi(\mu_{ij}) \propto 1$ [57] and the prior distribution becomes:

$$\pi\left(\mu_{ij}, \sigma^2_i\right) = \frac{1}{\sigma^2_{i1} - \sigma^2_{i0}}, \quad \sigma^2_{i0} < \sigma^2_i < \sigma^2_{i1}$$

Taking into account that the Normal likelihood function is

$$L\left(x_{i1}, \ldots, x_{in} | \mu_{ij}, \sigma^2_i\right) \propto \frac{1}{\sigma^n_i} \exp\left(-\frac{1}{2\sigma^2_i} \sum_{j=1}^{n} \left(x_{ij} - \mu_{ij}\right)^2\right)$$

where the symbol $\propto$ means "proportional to", the obtention of the posterior distribution is straightforward:

$$\pi\left(\mu_{ij}, \sigma^2_i | x_{ij}\right) \propto \frac{1}{\sigma^n_i} \exp\left(-\frac{1}{2\sigma^2_i} \sum_{j=1}^{n} \left(x_{ij} - \mu_{ij}\right)^2\right), \quad \sigma^2_{i0} < \sigma^2_i < \sigma^2_{i1}$$

From this posterior distribution, we may apply the Gibbs algorithm by obtaining samples from the one-dimensional conditional distributions, that is, from $\sigma^2_i | \mu_{ij,} x_{ij}$ and $\mu_{ij} | \mu_{i(j)}, \sigma^2_i, x_{ij}$ where $\mu_{i(j)} = \left(\mu_{i1}, \ldots, \mu_{ij-1}, \mu_{ij+1}, \ldots, \mu_{in}\right)$, because we obtain in both cases well-known distributions models and we are able to obtain simulated samples from them. For the variance, the posterior conditional distribution is:

$$\pi\left(\sigma^2_i | \mu_{ij}, x_{ij}\right) \propto \left(\frac{1}{\sigma^2_i}\right)^{n/2} \exp\left(-\frac{1}{2\sigma^2_i} \sum_{j=1}^{n} \left(x_{ij} - \mu_{ij}\right)^2\right),$$

$$\sigma^2_{i0} < \sigma^2_i < \sigma^2_{i1}$$

and therefore $\sigma^2_i | \mu_{ij}, x_{ij} \sim IG(\frac{n-2}{2}, \frac{1}{2}\sum_{j=1}^{n} (x_{ij} - \mu_{ij})^2)$ truncated at the interval $\sigma^2_{i0} < \sigma^2_i < \sigma^2_{i1}$ where $X \sim IG(\alpha, \beta)$ represents that X follows an Inverse-Gamma distribution with shape parameter α and scale parameter β whose density function is:

$$f(x) = \frac{\exp\left(-\frac{\beta}{x}\right)\left(\frac{\beta}{x}\right)^{\alpha}}{x\Gamma(\alpha)} \propto \exp\left(-\frac{\beta}{x}\right)\left(\frac{1}{x}\right)^{\alpha+1}$$

For the unidimensional mean, the posterior conditional distribution is:

$$\mu_{ij} | \mu_{i(j)}, \sigma^2_i, x_{ij} \sim N\left(x_{ij}, \sigma^2_i\right).$$

Applying the described Gibbs algorithm to inputs, good outputs and bad outputs, we obtain a large sample from the posterior distributions of μ^x_{ij}, σ^x_{ij}, μ^g_{ij}, σ^g_{ij}, μ^b_{ij}, σ^b_{ij}. Solving the deterministic equivalent problems, we obtain a large sample from the posterior distributions of the efficiencies of the DMUs and then we can estimate the efficiencies and the Monte Carlo Methods may be applied to retrieve the inferences about them.

Appendix B. RAW DEA results

B.1. Managerial DEA results.

	Architecture	Extractor	Resolution	# Proposals	mAP	mAP (large)	mAP (medium)	mAP (small)	mAP @. 75IOU	mAP @. 50IOU	# model params	Memory	Memory std	CPU	CPU std	GPU	GPU std	FLOPS	DEA result
1	Faster RCNN	VGG	300	300	22,9	37,7	16,4	2,4	22,9	41,1	13,851	1699	81	1286	27	201	16	6432	0,992,863,189
2	Faster RCNN	VGG	300	10	19,6	32,8	12,7	1,7	20,4	33,4	13,851	1703	30	1322	41	203	16	2912	1
3	Faster RCNN	VGG	300	20	21,5	35,7	14,8	2,1	21,9	37,6	13,851	1870	183	1261	5	204	15	3033	1
4	Faster RCNN	VGG	300	50	22,5	37	16,1	2,3	22,5	40,2	13,851	1842	258	1280	41	202	15	3397	0,994,721,065
5	Faster RCNN	VGG	300	100	22,9	37,5	16,4	2,5	22,8	41	13,851	1713	152	1272	8	201	17	4004	1
6	Faster RCNN	VGG	600	300	25,7	37,2	23	5,6	25,7	46	13,851	2048	210	1668	71	247	14	14,959	0,936,694,152
7	Faster RCNN	VGG	600	10	19	28,3	15,6	3,7	20,1	32,2	13,851	1853	388	1272	7	206	15	11,439	1
8	Faster RCNN	VGG	600	20	22,3	32,8	19,3	5	23,1	38,7	13,851	1984	182	1282	5	205	16	1156	1
9	Faster RCNN	VGG	600	50	24,8	35,6	22,2	5,9	25,3	44	13,851	1822	29	1327	6	209	16	11,925	1
10	Faster RCNN	VGG	600	100	25,6	36,5	22,9	6,2	25,8	45,6	13,851	1948	210	1384	11	217	15	12,532	1
11	Faster RCNN	Resnet 101	300	300	26,5	43,5	19	3	27,2	44,6	6315	3236	190	2484	28	206	14	23,942	0,691,020,311
12	Faster RCNN	Resnet 101	300	10	23,1	39,3	14,8	2,3	24,6	37,1	6315	560	19	355	13	92	13	2682	1
13	Faster RCNN	Resnet 101	300	20	24,8	41,6	17,1	2,6	26,1	40,9	6315	664	40	427	19	102	15	3415	0,990,104,601
14	Faster RCNN	Resnet 101	300	50	25,8	42,7	18,4	2,9	26,7	43,4	6315	1021	10	634	15	112	17	5614	1
15	Faster RCNN	Resnet 101	300	100	26,3	43,2	18,8	3,1	27	44,3	6315	1459	33	1029	12	136	14	92,8	0,902,364,702
16	Faster RCNN	Resnet 101	600	300	32,2	47,1	28,7	7,5	33,9	52,6	6315	4681	209	3127	17	248	14	29,661	1
17	Faster RCNN	Resnet 101	600	10	26,3	40	21,6	4,6	29,1	40,6	6315	1884	132	967	6	115	15	8402	1
18	Faster RCNN	Resnet 101	600	20	29,4	44,3	25,3	6,2	31,8	46,4	6315	1985	81	1085	93	126	16	9135	1
19	Faster RCNN	Resnet 101	600	50	31,3	46,1	27,6	7,2	33,4	50,6	6315	2267	106	1287	46	140	17	11,334	1
20	Faster RCNN	Resnet 101	600	100	32	46,7	28,5	7,5	33,9	52	6315	2756	138	1652	47	164	17	150	0,945,580,267
21	Faster RCNN	Resnet 101	300	300	29,5	46,1	23,4	5,2	31,1	48,3	6315	4271	262	2930	23	233	15	28,999	0,821,391,556
22	Faster RCNN	Resnet 101	300	10	24,5	41,1	16,6	3,5	27	37,9	6315	1378	44	789	23	108	13	7739	0,975,277,232
23	Faster RCNN	Resnet 101	300	20	26,9	43,9	19,9	4,3	29,1	42,7	6315	1464	49	875	10	120	16	8472	1
24	Faster RCNN	Resnet 101	300	50	28,7	45,4	22,3	4,9	30,6	46,5	6315	1703	42	1098	8	132	18	10,671	1
25	Faster RCNN	Resnet 101	300	100	29,2	45,8	23,1	5,2	30,9	47,8	6315	2218	162	1502	12	161	15	14,337	0,937,373,734
26	Faster RCNN	Resnet 101	600	300	34,4	47,7	31,8	10,7	37,1	54,6	6315	8043	301	4859	37	390	15	49,247	0,950,404,967
27	Faster RCNN	Resnet 101	600	10	26,7	40,7	22,2	5,1	29,6	40,2	6315	5056	35	2672	34	196	16	27,987	0,792,525,311
28	Faster RCNN	Resnet 101	600	20	30	44,6	26,8	6,9	33	46,2	6315	5146	143	2729	44	214	15	28,721	0,806,356,447
29	Faster RCNN	Resnet 101	600	50	32,7	46,6	29,9	8,9	35,5	51,2	6315	5375	79	3047	113	234	16	3092	0,834,281,018
30	Faster RCNN	Resnet 101	600	100	33,7	47,4	31,1	9,9	36,5	53,4	6315	5518	135	3384	65	267	16	34,585	0,861,698,229
31	Faster RCNN	Inception V2	300	300	15,4	27,4	7,7	0,5	14,8	29	1331	2370	51	1388	5	178	15	11,822	0,873,408,819
32	Faster RCNN	Inception V2	300	10	13,3	24,3	5,2	0,3	13,4	23,7	1331	211	6	153	7	79	13	7,63	0,952,521,136
33	Faster RCNN	Inception V2	300	20	14,3	26	6,3	0,3	14,1	26,3	1331	280	11	198	6	90	13	1144	0,831,436,862
34	Faster RCNN	Inception V2	300	50	15	26,9	7,2	0,5	14,5	28	1331	495	21	327	7	99	17	2288	0,764,144,374
35	Faster RCNN	Inception V2	300	100	15,3	27,3	7,4	0,5	14,7	28,5	1331	861	51	535	7	118	15	4195	0,709,397,299
36	Faster RCNN	Inception V2	600	300	21,9	34,4	17,6	3,1	21,8	39,5	1331	2596	63	1611	21	206	15	12,957	0,697,367,529
37	Faster RCNN	Inception V2	600	10	16,4	27,3	10,7	1,9	17,2	28	1331	572	11	383	7	82	12	1898	0,925,119,313
38	Faster RCNN	Inception V2	600	20	19	31,4	13,6	2,2	19,6	33,3	1331	669	16	446	22	98	13	2279	088,348,797
39	Faster RCNN	Inception V2	600	50	20,8	33,5	16,1	2,5	21	37,2	1331	871	17	574	22	116	15	3423	0,840,390,039
40	Faster RCNN	Inception V2	600	100	21,5	34	16,9	2,8	21,5	38,7	1331	1152	42	769	16	104	4	53,3	1
41	Faster RCNN	Inception V3	300	300	23,6	40,6	15,1	1,6	23,5	41,4	2627	5016	128	3214	18	353	14	3066	0,622,920,627
42	Faster RCNN	Inception V3	300	10	20,2	36,1	11,1	1,3	21,3	33,7	2627	337	5	252	9	101	13	1577	1
43	Faster RCNN	Inception V3	300	20	21,9	38,4	12,9	1,6	22,6	37,3	2627	487	13	395	30	109	13	25,8	0,892,216,604
44	Faster RCNN	Inception V3	300	50	23	39,8	14,4	1,7	23,2	40,1	2627	1017	40	669	11	137	13	5589	0,809,913,591
45	Faster RCNN	Inception V3	300	100	23,4	40,3	14,9	1,7	23,4	41	2627	1734	106	1188	22	181	14	10,603	0,670,253,151
46	Faster RCNN	Inception V3	600	300	29,6	44,9	25	5,3	30,6	50,2	2627	5386	143	3662	153	385	13	32,509	0,628,050,424
47	Faster RCNN	Inception V3	600	10	23,9	38,5	17,4	3,2	25,9	38,1	2627	799	71	525	18	116	12	3426	0,936,410,729
48	Faster RCNN	Inception V3	600	20	26,6	41,9	21	4,2	28,3	43,6	2627	1014	21	613	11	130	13	4429	0,959,575,307
49	Faster RCNN	Inception V3	600	50	28,6	44	23,5	4,9	29,9	47,9	2627	1522	14	1048	60	153	16	7437	0,901,848,556
50	Faster RCNN	Inception V3	600	100	29,2	44,6	24,3	5,4	30,2	49,4	2627	2293	81	1482	5	206	13	12,451	1
51	Faster RCNN	Inception Resnet V2	300	300	28,4	47	20,5	3,6	29,3	47,1	6002	14,769	103	7118	104	625	15	63,998	0,584,122,913
52	Faster RCNN	Inception Resnet V2	300	10	24,7	42,7	15,6	2,8	26,6	38,8	6002	903	14	518	14	130	14	3345	1

(*continued on next page*)

	Architecture	Extractor	Resolution	# Proposals	mAP	mAP (large)	mAP (medium)	mAP (small)	mAP @. 75IOU	mAP @. 50IOU	# model params	Memory	Memory std	CPU	CPU std	GPU	GPU std	FLOPS	DEA result
53	Faster RCNN	Inception Resnet V2	300	20	26,6	45	18,1	3,4	28,1	43	6002	1295	36	756	19	147	13	5436	08,609,229
54	Faster RCNN	Inception Resnet V2	300	50	27,9	46,3	19,9	3,7	29	45,9	6002	2622	69	1467	23	197	13	11,711	0,767,276,493
55	Faster RCNN	Inception Resnet V2	300	100	28,2	46,7	20,3	3,7	28,2	46,7	6002	4908	212	2670	16	283	13	22,168	0,731,036,561
56	Faster RCNN	Inception Resnet V2	600	300	35,4	52,8	31,1	8,3	37,5	56,3	6002	15,799	431	7652	31	671	13	68,043	0,920,287,958
57	Faster RCNN	Inception Resnet V2	600	10	29	46,1	22,9	5,4	31,8	43,5	6002	2173	57	1138	28	157	14	73,9	09,074,608
58	Faster RCNN	Inception Resnet V2	600	20	32,2	49,6	27,1	6,7	34,9	49,6	6002	2651	50	1297	33	177	14	9481	0,923,089,298
59	Faster RCNN	Inception Resnet V2	600	50	34,2	51,5	29,7	7,8	36,6	54,1	6002	4172	119	2058	62	232	13	15,756	0,816,011,426
60	Faster RCNN	Inception Resnet V2	600	100	34,9	52,4	30,5	8,1	37,1	55,4	6002	6222	257	3390	47	319	14	26,213	0,792,351,318
61	Faster RCNN	Inception Resnet V2	300	300	29,8	48,4	22,8	4,5	30,9	49,2	6002	6053	267	2905	66	272	19	66,545	0,638,583,584
62	Faster RCNN	Inception Resnet V2	300	10	25,1	43,5	16,2	2,7	27,2	39	6002	6207	79	2937	31	287	15	5892	0,627,859,672
63	Faster RCNN	Inception Resnet V2	300	20	27,4	46,2	19,3	3,8	29,4	43,7	6002	6138	109	2917	24	286	16	7984	0,663,382,971
64	Faster RCNN	Inception Resnet V2	300	50	28,9	47,7	21,3	4,4	30,4	47,1	6002	5920	177	3036	15	283	16	14,258	0,726,514,381
65	Faster RCNN	Inception Resnet V2	300	100	29,5	48,1	22,1	4,5	30,7	48,5	6002	6141	81	2945	52	281	17	24,715	0,629,430,321
66	Faster RCNN	Inception Resnet V2	600	300	35,7	52,5	32	8,9	38	56,5	6002	19,467	284	10,018	491	855	14	78,899	0,704,720,088
67	Faster RCNN	Inception Resnet V2	600	10	29,1	46,5	23,6	4,9	31,9	43,3	6002	6008	111	2889	34	285	15	18,245	0,664,052,088
68	Faster RCNN	Inception Resnet V2	600	20	32,2	49,8	27,5	6,6	34,9	49,2	6002	6536	186	3230	92	308	13	20,337	0,674,861,576
69	Faster RCNN	Inception Resnet V2	600	50	34,4	51,6	30,4	7,9	36,9	53,8	6002	7854	19	3849	29	367	15	26,611	0,992,603,286
70	Faster RCNN	Inception Resnet V2	600	100	35,3	52,1	31,5	8,5	37,6	55,6	6002	10,341	296	5132	38	465	14	37,069	080,771,235
71	Faster RCNN	MobileNet	300	300	16,4	27,8	9,9	1	15,5	31,5	6,06	1147	28	427	22	94	17	2523	0,876,872,748
72	Faster RCNN	MobileNet	300	10	14,4	25,3	7,3	0,8	14,2	26,5	6,06	143	5	100	5	54	7	2,55	1
73	Faster RCNN	MobileNet	300	20	15,5	27	8,6	0,9	15	29,3	6,06	172	6	123	4	61	9	3,33	1
74	Faster RCNN	MobileNet	300	50	16,2	27,7	9,5	1	15,3	31	6,06	284	4	146	4	72	14	5,67	1
75	Faster RCNN	MobileNet	300	100	16,4	27,9	9,9	0,9	15,5	31,5	6,06	447	17	203	9	81	14	9,59	0,876,768,764
76	Faster RCNN	MobileNet	600	300	19,8	29,3	17,5	3,8	18,3	38,5	6,06	1412	68	610	8	121	16	30,5	1
77	Faster RCNN	MobileNet	600	10	15,1	23,7	11,8	2,4	14,8	28	6,06	487	7	273	12	63	6	7,82	1
78	Faster RCNN	MobileNet	600	20	17,4	26,8	14,4	2,8	16,4	33,2	6,06	507	4	292	2	69	9	8,6	1
79	Faster RCNN	MobileNet	600	50	19,1	28,6	16,7	3,5	17,8	37	6,06	572	25	349	11	79	14	1094	1
80	Faster RCNN	MobileNet	600	100	19,7	29,2	17,4	3,7	18,1	38,3	6,06	768	10	376	18	98	14	1485	1
81	R-FCN	Resnet 101	300	300	25,2	41,1	18,7	2,5	25,9	43,4	6848	992	30	444	15	103	6	2714	1
82	R-FCN	Resnet 101	300	100	25,2	41,2	18,6	2,6	26	43,5	6848	861	47	437	25	94	5	2714	1
83	R-FCN	Resnet 101	300	50	24,9	40,9	1821	2,5	25,8	42,7	6848	798	51	425	20	92	6	2714	1
84	R-FCN	Resnet 101	300	20	23,8	39,7	16,8	2,1	25,2	40,1	6848	713	79	413	10	90	6	2714	1
85	R-FCN	Resnet 101	300	10	22,1	37,6	14,4	1,8	23,9	36,3	6848	708	25	428	31	89	6	2714	1
86	R-FCN	Resnet 101	600	300	30,4	43,3	28,3	7,2	32	51,4	6848	2878	75	1536	42	161	8	1073	1
87	R-FCN	Resnet 101	600	100	30,5	43,4	28,5	7,4	32,1	51,6	6848	2651	19	1465	62	141	9	1073	1
88	R-FCN	Resnet 101	600	50	30,1	43,3	27,9	7,3	31,8	50,4	6848	2591	100	1419	40	134	8	1073	1
89	R-FCN	Resnet 101	600	20	28,3	41,3	25,7	6,2	30,4	46,5	6848	2600	69	1503	103	131	8	1073	1
90	R-FCN	Resnet 101	600	10	25,3	37,9	21,9	4,9	27,7	40,5	6848	2530	34	1438	64	131	8	1073	0,998,772,809
91	R-FCN	Resnet 101	300	300	27,6	43,3	22,7	4,4	29	46,1	6848	2190	26	1330	25	147	7	10,067	1
92	R-FCN	Resnet 101	300	100	27,4	43,5	22,3	4,2	28,9	45,7	6848	2101	97	1314	21	133	6	10,067	0,970,523,273
93	R-FCN	Resnet 101	300	50	27	43,3	21,6	3,7	28,7	44,5	6848	2072	86	1325	57	128	6	10,067	0,947,762,286
94	R-FCN	Resnet 101	300	20	25,2	42	18,6	3	27,4	40,7	6848	2088	86	1303	23	127	6	10,067	0,927,504,965
95	R-FCN	Resnet 101	300	10	23,1	39,4	15,7	2,4	25,5	36,5	6848	1943	93	1326	57	126	6	10,067	0,885,004,763
96	R-FCN	Resnet 101	600	300	31,9	43	30,8	10,4	33,7	52,4	6848	7746	394	4823	145	389	14	39,178	084,314,076
97	R-FCN	Resnet 101	600	100	31,7	43,2	30,7	10	33,8	52,2	6848	7143	97	4747	125	327	13	39,178	085,856,585
98	R-FCN	Resnet 101	600	50	31,1	42,9	30,1	9,3	33,3	50,8	6848	7842	163	4769	108	313	12	39,178	0,849,016,854
99	R-FCN	Resnet 101	600	20	28,8	41,7	26,9	7,2	31,5	46,1	6848	7562	251	4801	134	304	13	39,178	0,704,610,835
100	R-FCN	Resnet 101	600	10	25,6	38,8	22,6	5,4	28,3	40,1	6848	7553	244	4685	113	302	13	39,178	0,591,162,527
101	R-FCN	Inception V2	300	300	15,4	27,8	7	0	14,5	29,5	1806	286	14	144	7	72	5	5,09	1
102	R-FCN	Inception V2	300	100	15,5	27,9	7,2	1	14,6	29,9	1806	208	11	125	6	69	6	5,09	1
103	R-FCN	Inception V2	300	50	15,3	27,7	7,1	1	14,5	29,6	1806	202	13	125	6	66	6	5,09	1

(*continued on next page*)

	Architecture	Extractor	Resolution	# Proposals	mAP	mAP (large)	mAP (medium)	mAP (small)	mAP @. 75IOU	mAP @. 50IOU	# model params	Memory	Memory std	CPU	CPU std	GPU	GPU std	FLOPS	DEA result
104	R-FCN	Inception V2	300	20	14,7	26,8	6,5	0	14,1	27,9	1806	177	10	123	8	64	6	5,09	1
105	R-FCN	Inception V2	300	10	13,8	25,2	5,7	0	13,6	25,6	1806	164	12	121	7	66	6	5,09	1
106	R-FCN	Inception V2	600	300	19,8	31,8	15,2	1,8	18,2	38,3	1806	715	6	445	16	101	6	1985	1
107	R-FCN	Inception V2	600	100	20,1	31,9	15,6	2	18,5	39,1	1806	647	19	394	15	87	6	1985	1
108	R-FCN	Inception V2	600	50	19,7	31,6	15	2	18,3	38,1	1806	624	25	391	19	84	7	1985	0,974,865,874
109	R-FCN	Inception V2	600	20	18,3	30,1	13,2	1,8	17,4	34,7	1806	609	9	378	5	81	6	1985	1
110	R-FCN	Inception V2	600	10	15,9	26,5	10,6	1,6	15,6	29,5	1806	581	12	376	8	79	7	1985	0,977,454,298
111	R-FCN	Inception Resnet V2	300	300	22,5	39,6	13,2	1	22,7	39,8	6506	696	108	339	10	108	6	1499	1
112	R-FCN	Inception Resnet V2	300	100	22,8	39,8	13,8	1,1	23	40,7	6506	642	71	315	13	100	7	1499	1
113	R-FCN	Inception Resnet V2	300	50	22,8	39,6	14	1,2	23	40,6	6506	679	33	314	7	101	6	1499	1
114	R-FCN	Inception Resnet V2	300	20	22,3	38,9	13,5	1	22,8	39,1	6506	611	42	312	13	99	6	1499	1
115	R-FCN	Inception Resnet V2	300	10	21,1	37,5	12,1	1,1	22,1	36	6506	637	96	324	10	98	7	1499	1
116	R-FCN	Inception Resnet V2	600	300	30,7	46,2	26,8	5,6	31,7	52,3	6506	2173	108	1048	24	165	6	6396	1
117	R-FCN	Inception Resnet V2	600	100	30,9	46,4	27	6,1	31,9	52,8	6506	2121	36	993	26	150	6	6396	1
118	R-FCN	Inception Resnet V2	600	50	30,7	46,1	26,8	6,2	32	52,3	6506	2150	83	1060	41	146	6	6396	1
119	R-FCN	Inception Resnet V2	600	20	29,3	44,9	25	5,4	31,1	48,8	6506	2051	12	1007	30	144	6	6396	1
120	R-FCN	Inception Resnet V2	600	10	27,1	42,7	22,1	4,4	29,3	43,9	6506	2151	62	1034	12	141	5	6396	1
121	R-FCN	Inception Resnet V2	300	300	26	43,5	18,7	2	26,4	45,3	6506	1609	37	827	14	142	7	4774	0,967,072,717
122	R-FCN	Inception Resnet V2	300	100	26	43,5	18,6	2,1	26,4	45,2	6506	1557	28	841	10	134	6	4774	1
123	R-FCN	Inception Resnet V2	300	50	25,6	43,2	17,9	1,9	26,1	44,3	6506	1564	14	808	5	133	7	4774	1
124	R-FCN	Inception Resnet V2	300	20	24,4	42,4	15,9	1,7	25,4	41,3	6506	1499	83	796	3	128	8	4774	1
125	R-FCN	Inception Resnet V2	300	10	22,5	40,1	13,5	1,5	23,9	37,4	6506	1501	102	791	14	131	6	4774	0,931,535,439
126	R-FCN	Inception Resnet V2	600	300	32,3	46,9	29,3	8	33,8	53,9	6506	7289	46	3220	60	368	8	20,545	1
127	R-FCN	Inception Resnet V2	600	100	32,1	46,8	29,3	8	33,7	53,7	6506	6653	53	3319	70	328	6	20,545	1
128	R-FCN	Inception Resnet V2	600	50	31,5	46,4	28,6	7,6	33,3	52,2	6506	6771	258	3315	32	313	7	20,545	1
129	R-FCN	Inception Resnet V2	600	20	29,7	45,2	26,4	6,7	32	48,3	6506	6958	137	3318	100	302	7	20,545	0,897,209,934
130	R-FCN	Inception Resnet V2	600	10	27,1	42,7	22,7	4,8	29,7	42,9	6506	7001	40	3320	21	299	7	20,545	0,934,269,357
131	R-FCN	MobileNet	300	300	15	26,3	7,6	0	13,7	29,4	10,8	246	4	114	4	66	5	2,44	1
132	R-FCN	MobileNet	300	100	15,2	26,5	7,9	1	13,8	30	10,8	170	10	94	8	57	7	2,44	1
133	R-FCN	MobileNet	300	50	15,2	26,5	7,8	1	13,8	30	10,8	159	10	92	8	60	5	2,44	1
134	R-FCN	MobileNet	300	20	14,7	25,9	7,3	1	13,6	28,7	10,8	150	10	96	7	59	5	2,44	1
135	R-FCN	MobileNet	300	10	13,8	24,4	6,5	1	13,1	26,4	10,8	137	11	97	5	58	5	2,44	1
136	R-FCN	MobileNet	600	300	16,5	25,8	13,2	1,5	14	33,8	10,8	607	17	341	25	86	5	9,49	1
137	R-FCN	MobileNet	600	100	17	26,2	13,8	1,8	14,3	35,1	10,8	537	6	296	8	73	6	9,49	1
138	R-FCN	MobileNet	600	50	16,9	26	13,7	2	14,2	34,8	10,8	514	2	283	8	70	6	9,49	1
139	R-FCN	MobileNet	600	20	15,9	24,9	12,6	1,8	13,7	32,3	10,8	511	15	285	8	69	5	9,49	1
140	R-FCN	MobileNet	600	10	14,1	22,2	10,6	1,3	12,5	27,9	10,8	507	20	280	9	67	5	9,49	1

B.2. Probabilistic DEA results.

	Architecture	Extractor	Resolution	# Proposals	mAP	mAP (large)	mAP (medium)	mAP (small)	mAP @. 75IOU	mAP @. 50IOU	# model params	Memory	Memory std	CPU	CPU std	GPU	GPU std	FLOPS	DEA result
1	Faster RCNN	VGG	300	300	22,9	37,7	16,4	2,4	22,9	41,1	13,851	1699	81	1286	27	201	16	6432	1
2	Faster RCNN	VGG	300	10	19,6	32,8	12,7	1,7	20,4	33,4	13,851	1703	30	1322	41	203	16	2912	1
3	Faster RCNN	VGG	300	20	21,5	35,7	14,8	2,1	21,9	37,6	13,851	1870	183	1261	5	204	15	3033	1
4	Faster RCNN	VGG	300	50	22,5	37	16,1	2,3	22,5	40,2	13,851	1842	258	1280	41	202	15	3397	1
5	Faster RCNN	VGG	300	100	22,9	37,5	16,4	2,5	22,8	41	13,851	1713	152	1272	8	201	17	4004	1
6	Faster RCNN	VGG	600	300	25,7	37,2	23	5,6	25,7	46	13,851	2048	210	1668	71	247	14	14,959	1
7	Faster RCNN	VGG	600	10	19	28,3	15,6	3,7	20,1	32,2	13,851	1853	388	1272	7	206	15	11,439	1
8	Faster RCNN	VGG	600	20	22,3	32,8	19,3	5	23,1	38,7	13,851	1984	182	1282	5	205	16	1156	1
9	Faster RCNN	VGG	600	50	24,8	35,6	22,2	5,9	25,3	44	13,851	1822	29	1327	6	209	16	11,925	1
10	Faster RCNN	VGG	600	100	25,6	36,5	22,9	6,2	25,8	45,6	13,851	1948	210	1384	11	217	15	12,532	1
11	Faster RCNN	Resnet 101	300	300	26,5	43,5	19	3	27,2	44,6	6315	3236	190	2484	28	206	14	23,942	0,721,635,676
12	Faster RCNN	Resnet 101	300	10	23,1	39,3	14,8	2,3	24,6	37,1	6315	560	19	355	13	92	13	2682	1
13	Faster RCNN	Resnet 101	300	20	24,8	41,6	17,1	2,6	26,1	40,9	6315	664	40	427	19	102	15	3415	1
14	Faster RCNN	Resnet 101	300	50	25,8	42,7	18,4	2,9	26,7	43,4	6315	1021	10	634	15	112	17	5614	1
15	Faster RCNN	Resnet 101	300	100	26,3	43,2	18,8	3,1	27	44,3	6315	1459	33	1029	12	136	14	92,8	0,932,975,719
16	Faster RCNN	Resnet 101	600	300	32,2	47,1	28,7	7,5	33,9	52,6	6315	4681	209	3127	17	248	14	29,661	1
17	Faster RCNN	Resnet 101	600	10	26,3	40	21,6	4,6	29,1	40,6	6315	1884	132	967	6	115	15	8402	1
18	Faster RCNN	Resnet 101	600	20	29,4	44,3	25,3	6,2	31,8	46,4	6315	1985	81	1085	93	126	16	9135	1
19	Faster RCNN	Resnet 101	600	50	31,3	46,1	27,6	7,2	33,4	50,6	6315	2267	106	1287	46	140	17	11,334	1
20	Faster RCNN	Resnet 101	600	100	32	46,7	28,5	7,5	33,9	52	6315	2756	138	1652	47	164	17	150	097,035,316
21	Faster RCNN	Resnet 101	300	300	29,5	46,1	23,4	5,2	31,1	48,3	6315	4271	262	2930	23	233	15	28,999	0,856,038,972
22	Faster RCNN	Resnet 101	300	10	24,5	41,1	16,6	3,5	27	37,9	6315	1378	44	789	23	108	13	7739	0,979,869,385
23	Faster RCNN	Resnet 101	300	20	26,9	43,9	19,9	4,3	29,1	42,7	6315	1464	49	875	10	120	16	8472	1
24	Faster RCNN	Resnet 101	300	50	28,7	45,4	22,3	4,9	30,6	46,5	6315	1703	42	1098	8	132	18	10,671	1
25	Faster RCNN	Resnet 101	300	100	29,2	45,8	23,1	5,2	30,9	47,8	6315	2218	162	1502	12	161	15	14,337	097,948,059
26	Faster RCNN	Resnet 101	600	300	34,4	47,7	31,8	10,7	37,1	54,6	6315	8043	301	4859	37	390	15	49,247	1
27	Faster RCNN	Resnet 101	600	10	26,7	40,7	22,2	5,1	29,6	40,2	6315	5056	35	2672	34	196	16	27,987	0,804,383,004
28	Faster RCNN	Resnet 101	600	20	30	44,6	26,8	6,9	33	46,2	6315	5146	143	2729	44	214	15	28,721	0,837,371,399
29	Faster RCNN	Resnet 101	600	50	32,7	46,6	29,9	8,9	35,5	51,2	6315	5375	79	3047	113	234	16	3092	0,940,473,942
30	Faster RCNN	Resnet 101	600	100	33,7	47,4	31,1	9,9	36,5	53,4	6315	5518	135	3384	65	267	16	34,585	1
31	Faster RCNN	Inception V2	300	300	15,4	27,4	7,7	0,5	14,8	29	1331	2370	51	1388	5	178	15	11,822	0,878,056,912
32	Faster RCNN	Inception V2	300	10	13,3	24,3	5,2	0,3	13,4	23,7	1331	211	6	153	7	79	13	7,63	0,969,136,168
33	Faster RCNN	Inception V2	300	20	14,3	26	6,3	0,3	14,1	26,3	1331	280	11	198	6	90	13	1144	0,849,455,864
34	Faster RCNN	Inception V2	300	50	15	26,9	7,2	0,5	14,5	28	1331	495	21	327	7	99	17	2288	0,770,753,102
35	Faster RCNN	Inception V2	300	100	15,3	27,3	7,4	0,5	14,7	28,5	1331	861	51	535	7	118	15	4195	0,711,132,149
36	Faster RCNN	Inception V2	600	300	21,9	34,4	17,6	3,1	21,8	39,5	1331	2596	63	1611	21	206	15	12,957	0,714,178,401
37	Faster RCNN	Inception V2	600	10	16,4	27,3	10,7	1,9	17,2	28	1331	572	11	383	7	82	12	1898	0,930,786,594
38	Faster RCNN	Inception V2	600	20	19	31,4	13,6	2,2	19,6	33,3	1331	669	16	446	22	98	13	2279	0,918,079,095
39	Faster RCNN	Inception V2	600	50	20,8	33,5	16,1	2,5	21	37,2	1331	871	17	574	22	116	15	3423	0,856,791,909
40	Faster RCNN	Inception V2	600	100	21,5	34	16,9	2,8	21,5	38,7	1331	1152	42	769	16	104	4	53,3	1
41	Faster RCNN	Inception V3	300	300	23,6	40,6	15,1	1,6	23,5	41,4	2627	5016	128	3214	18	353	14	3066	0,653,800,701
42	Faster RCNN	Inception V3	300	10	20,2	36,1	11,1	1,3	21,3	33,7	2627	337	5	252	9	101	13	1577	1
43	Faster RCNN	Inception V3	300	20	21,9	38,4	12,9	1,6	22,6	37,3	2627	487	13	395	30	109	13	25,8	0,997,415,181
44	Faster RCNN	Inception V3	300	50	23	39,8	14,4	1,7	23,2	40,1	2627	1017	40	669	11	137	13	5589	0,851,708,909
45	Faster RCNN	Inception V3	300	100	23,4	40,3	14,9	1,7	23,4	41	2627	1734	106	1188	22	181	14	10,603	0,690,446,077
46	Faster RCNN	Inception V3	600	300	29,6	44,9	25	5,3	30,6	50,2	2627	5386	143	3662	153	385	13	32,509	0,628,652,862
47	Faster RCNN	Inception V3	600	10	23,9	38,5	17,4	3,2	25,9	38,1	2627	799	71	525	18	116	12	3426	1
48	Faster RCNN	Inception V3	600	20	26,6	41,9	21	4,2	28,3	43,6	2627	1014	21	613	11	130	13	4429	1
49	Faster RCNN	Inception V3	600	50	28,6	44	23,5	4,9	29,9	47,9	2627	1522	14	1048	60	153	16	7437	1
50	Faster RCNN	Inception V3	600	100	29,2	44,6	24,3	5,4	30,2	49,4	2627	2293	81	1482	5	206	13	12,451	1
51	Faster RCNN	Inception Resnet V2	300	300	28,4	47	20,5	3,6	29,3	47,1	6002	14,769	103	7118	104	625	15	63,998	0,652,455,535
52	Faster RCNN	Inception Resnet V2	300	10	24,7	42,7	15,6	2,8	26,6	38,8	6002	903	14	518	14	130	14	3345	1

(*continued on next page*)

	Architecture	Extractor	Resolution	# Proposals	mAP	mAP (large)	mAP (medium)	mAP (small)	mAP @. 75IOU	mAP @. 50IOU	# model params	Memory	Memory std	CPU	CPU std	GPU	GPU std	FLOPS	DEA result
53	Faster RCNN	Inception Resnet V2	300	20	26,6	45	18,1	3,4	28,1	43	6002	1295	36	756	19	147	13	5436	0,932,407,582
54	Faster RCNN	Inception Resnet V2	300	50	27,9	46,3	19,9	3,7	29	45,9	6002	2622	69	1467	23	197	13	11,711	0,820,250,506
55	Faster RCNN	Inception Resnet V2	300	100	28,2	46,7	20,3	3,7	28,2	46,7	6002	4908	212	2670	16	283	13	22,168	0,842,905,455
56	Faster RCNN	Inception Resnet V2	600	300	35,4	52,8	31,1	8,3	37,5	56,3	6002	15,799	431	7652	31	671	13	68,043	1
57	Faster RCNN	Inception Resnet V2	600	10	29	46,1	22,9	5,4	31,8	43,5	6002	2173	57	1138	28	157	14	73,9	0,947,403,184
58	Faster RCNN	Inception Resnet V2	600	20	32,2	49,6	27,1	6,7	34,9	49,6	6002	2651	50	1297	33	177	14	9481	0,971,769,717
59	Faster RCNN	Inception Resnet V2	600	50	34,2	51,5	29,7	7,8	36,6	54,1	6002	4172	119	2058	62	232	13	15,756	0,938,547,812
60	Faster RCNN	Inception Resnet V2	600	100	34,9	52,4	30,5	8,1	37,1	55,4	6002	6222	257	3390	47	319	14	26,213	0,916,369,468
61	Faster RCNN	Inception Resnet V2	300	300	29,8	48,4	22,8	4,5	30,9	49,2	6002	6053	267	2905	66	272	19	66,545	0,680,231,761
62	Faster RCNN	Inception Resnet V2	300	10	25,1	43,5	16,2	2,7	27,2	39	6002	6207	79	2937	31	287	15	5892	0,690,971,126
63	Faster RCNN	Inception Resnet V2	300	20	27,4	46,2	19,3	3,8	29,4	43,7	6002	6138	109	2917	24	286	16	7984	0,781,504,982
64	Faster RCNN	Inception Resnet V2	300	50	28,9	47,7	21,3	4,4	30,4	47,1	6002	5920	177	3036	15	283	16	14,258	0,838,422,855
65	Faster RCNN	Inception Resnet V2	300	100	29,5	48,1	22,1	4,5	30,7	48,5	6002	6141	81	2945	52	281	17	24,715	0,673,962,641
66	Faster RCNN	Inception Resnet V2	600	300	35,7	52,5	32	8,9	38	56,5	6002	19,467	284	10,018	491	855	14	78,899	0,912,968,467
67	Faster RCNN	Inception Resnet V2	600	10	29,1	46,5	23,6	4,9	31,9	43,3	6002	6008	111	2889	34	285	15	18,245	0,696,804,143
68	Faster RCNN	Inception Resnet V2	600	20	32,2	49,8	27,5	6,6	34,9	49,2	6002	6536	186	3230	92	308	13	20,337	0,794,358,534
69	Faster RCNN	Inception Resnet V2	600	50	34,4	51,6	30,4	7,9	36,9	53,8	6002	7854	19	3849	29	367	15	26,611	1
70	Faster RCNN	Inception Resnet V2	600	100	35,3	52,1	31,5	8,5	37,6	55,6	6002	10,341	296	5132	38	465	14	37,069	0,960,959,275
71	Faster RCNN	MobileNet	300	300	16,4	27,8	9,9	1	15,5	31,5	6,06	1147	28	427	22	94	17	2523	0,889,689,454
72	Faster RCNN	MobileNet	300	10	14,4	25,3	7,3	0,8	14,2	26,5	6,06	143	5	100	5	54	7	2,55	1
73	Faster RCNN	MobileNet	300	20	15,5	27	8,6	0,9	15	29,3	6,06	172	6	123	4	61	9	3,33	1
74	Faster RCNN	MobileNet	300	50	16,2	27,7	9,5	1	15,3	31	6,06	284	4	146	4	72	14	5,67	1
75	Faster RCNN	MobileNet	300	100	16,4	27,9	9,9	0,9	15,5	31,5	6,06	447	17	203	9	81	14	9,59	087,861,171
76	Faster RCNN	MobileNet	600	300	19,8	29,3	17,5	3,8	18,3	38,5	6,06	1412	68	610	8	121	16	30,5	1
77	Faster RCNN	MobileNet	600	10	15,1	23,7	11,8	2,4	14,8	28	6,06	487	7	273	12	63	6	7,82	1
78	Faster RCNN	MobileNet	600	20	17,4	26,8	14,4	2,8	16,4	33,2	6,06	507	4	292	2	69	9	8,6	1
79	Faster RCNN	MobileNet	600	50	19,1	28,6	16,7	3,5	17,8	37	6,06	572	25	349	11	79	14	1094	1
80	Faster RCNN	MobileNet	600	100	19,7	29,2	17,4	3,7	18,1	38,3	6,06	768	10	376	18	98	14	1485	1
81	R-FCN	Resnet 101	300	300	25,2	41,1	18,7	2,5	25,9	43,4	6848	992	30	444	15	103	6	2714	1
82	R-FCN	Resnet 101	300	100	25,2	41,2	18,6	2,6	26	43,5	6848	861	47	437	25	94	5	2714	1
83	R-FCN	Resnet 101	300	50	24,9	40,9	1821	2,5	25,8	42,7	6848	798	51	425	20	92	6	2714	1
84	R-FCN	Resnet 101	300	20	23,8	39,7	16,8	2,1	25,2	40,1	6848	713	79	413	10	90	6	2714	1
85	R-FCN	Resnet 101	300	10	22,1	37,6	14,4	1,8	23,9	36,3	6848	708	25	428	31	89	6	2714	1
86	R-FCN	Resnet 101	600	300	30,4	43,3	28,3	7,2	32	51,4	6848	2878	75	1536	42	161	8	1073	1
87	R-FCN	Resnet 101	600	100	30,5	43,4	28,5	7,4	32,1	51,6	6848	2651	19	1465	62	141	9	1073	1
88	R-FCN	Resnet 101	600	50	30,1	43,3	27,9	7,3	31,8	50,4	6848	2591	100	1419	40	134	8	1073	1
89	R-FCN	Resnet 101	600	20	28,3	41,3	25,7	6,2	30,4	46,5	6848	2600	69	1503	103	131	8	1073	1
90	R-FCN	Resnet 101	600	10	25,3	37,9	21,9	4,9	27,7	40,5	6848	2530	34	1438	64	131	8	1073	1
91	R-FCN	Resnet 101	300	300	27,6	43,3	22,7	4,4	29	46,1	6848	2190	26	1330	25	147	7	10,067	1
92	R-FCN	Resnet 101	300	100	27,4	43,5	22,3	4,2	28,9	45,7	6848	2101	97	1314	21	133	6	10,067	0,976,361,082
93	R-FCN	Resnet 101	300	50	27	43,3	21,6	3,7	28,7	44,5	6848	2072	86	1325	57	128	6	10,067	0,955,884,039
94	R-FCN	Resnet 101	300	20	25,2	42	18,6	3	27,4	40,7	6848	2088	86	1303	23	127	6	10,067	0,934,326,435
95	R-FCN	Resnet 101	300	10	23,1	39,4	15,7	2,4	25,5	36,5	6848	1943	93	1326	57	126	6	10,067	0,897,568,909
96	R-FCN	Resnet 101	600	300	31,9	43	30,8	10,4	33,7	52,4	6848	7746	394	4823	145	389	14	39,178	1
97	R-FCN	Resnet 101	600	100	31,7	43,2	30,7	10	33,8	52,2	6848	7143	97	4747	125	327	13	39,178	1
98	R-FCN	Resnet 101	600	50	31,1	42,9	30,1	9,3	33,3	50,8	6848	7842	163	4769	108	313	12	39,178	0,971,004,451
99	R-FCN	Resnet 101	600	20	28,8	41,7	26,9	7,2	31,5	46,1	6848	7562	251	4801	134	304	13	39,178	0,716,476,904
100	R-FCN	Resnet 101	600	10	25,6	38,8	22,6	5,4	28,3	40,1	6848	7553	244	4685	113	302	13	39,178	061,047,968
101	R-FCN	Inception V2	300	300	15,4	27,8	7	0	14,5	29,5	1806	286	14	144	7	72	5	5,09	1
102	R-FCN	Inception V2	300	100	15,5	27,9	7,2	1	14,6	29,9	1806	208	11	125	6	69	6	5,09	1
103	R-FCN	Inception V2	300	50	15,3	27,7	7,1	1	14,5	29,6	1806	202	13	125	6	66	6	5,09	1

(*continued on next page*)

	Architecture	Extractor	Resolution	# Proposals	mAP	mAP (large)	mAP (medium)	mAP (small)	mAP @. 75IOU	mAP @. 50IOU	# model params	Memory	Memory std	CPU	CPU std	GPU	GPU std	FLOPS	DEA result
104	R-FCN	Inception V2	300	20	14,7	26,8	6,5	0	14,1	27,9	1806	177	10	123	8	64	6	5,09	1
105	R-FCN	Inception V2	300	10	13,8	25,2	5,7	0	13,6	25,6	1806	164	12	121	7	66	6	5,09	1
106	R-FCN	Inception V2	600	300	19,8	31,8	15,2	1,8	18,2	38,3	1806	715	6	445	16	101	6	1985	1
107	R-FCN	Inception V2	600	100	20,1	31,9	15,6	2	18,5	39,1	1806	647	19	394	15	87	6	1985	1
108	R-FCN	Inception V2	600	50	19,7	31,6	15	2	18,3	38,1	1806	624	25	391	19	84	7	1985	1
109	R-FCN	Inception V2	600	20	18,3	30,1	13,2	1,8	17,4	34,7	1806	609	9	378	5	81	6	1985	1
110	R-FCN	Inception V2	600	10	15,9	26,5	10,6	1,6	15,6	29,5	1806	581	12	376	8	79	7	1985	0,994,897,448
111	R-FCN	Inception Resnet V2	300	300	22,5	39,6	13,2	1	22,7	39,8	6506	696	108	339	10	108	6	1499	1
112	R-FCN	Inception Resnet V2	300	100	22,8	39,8	13,8	1,1	23	40,7	6506	642	71	315	13	100	7	1499	1
113	R-FCN	Inception Resnet V2	300	50	22,8	39,6	14	1,2	23	40,6	6506	679	33	314	7	101	6	1499	1
114	R-FCN	Inception Resnet V2	300	20	22,3	38,9	13,5	1	22,8	39,1	6506	611	42	312	13	99	6	1499	1
115	R-FCN	Inception Resnet V2	300	10	21,1	37,5	12,1	1,1	22,1	36	6506	637	96	324	10	98	7	1499	1
116	R-FCN	Inception Resnet V2	600	300	30,7	46,2	26,8	5,6	31,7	52,3	6506	2173	108	1048	24	165	6	6396	1
117	R-FCN	Inception Resnet V2	600	100	30,9	46,4	27	6,1	31,9	52,8	6506	2121	36	993	26	150	6	6396	1
118	R-FCN	Inception Resnet V2	600	50	30,7	46,1	26,8	6,2	32	52,3	6506	2150	83	1060	41	146	6	6396	1
119	R-FCN	Inception Resnet V2	600	20	29,3	44,9	25	5,4	31,1	48,8	6506	2051	12	1007	30	144	6	6396	1
120	R-FCN	Inception Resnet V2	600	10	27,1	42,7	22,1	4,4	29,3	43,9	6506	2151	62	1034	12	141	5	6396	1
121	R-FCN	Inception Resnet V2	300	300	26	43,5	18,7	2	26,4	45,3	6506	1609	37	827	14	142	7	4774	1
122	R-FCN	Inception Resnet V2	300	100	26	43,5	18,6	2,1	26,4	45,2	6506	1557	28	841	10	134	6	4774	1
123	R-FCN	Inception Resnet V2	300	50	25,6	43,2	17,9	1,9	26,1	44,3	6506	1564	14	808	5	133	7	4774	1
124	R-FCN	Inception Resnet V2	300	20	24,4	42,4	15,9	1,7	25,4	41,3	6506	1499	83	796	3	128	8	4774	1
125	R-FCN	Inception Resnet V2	300	10	22,5	40,1	13,5	1,5	23,9	37,4	6506	1501	102	791	14	131	6	4774	0,937,080,879
126	R-FCN	Inception Resnet V2	600	300	32,3	46,9	29,3	8	33,8	53,9	6506	7289	46	3220	60	368	8	20,545	1
127	R-FCN	Inception Resnet V2	600	100	32,1	46,8	29,3	8	33,7	53,7	6506	6653	53	3319	70	328	6	20,545	1
128	R-FCN	Inception Resnet V2	600	50	31,5	46,4	28,6	7,6	33,3	52,2	6506	6771	258	3315	32	313	7	20,545	1
129	R-FCN	Inception Resnet V2	600	20	29,7	45,2	26,4	6,7	32	48,3	6506	6958	137	3318	100	302	7	20,545	0,898,411,248
130	R-FCN	Inception Resnet V2	600	10	27,1	42,7	22,7	4,8	29,7	42,9	6506	7001	40	3320	21	299	7	20,545	0,941,140,671
131	R-FCN	MobileNet	300	300	15	26,3	7,6	0	13,7	29,4	10,8	246	4	114	4	66	5	2,44	1
132	R-FCN	MobileNet	300	100	15,2	26,5	7,9	1	13,8	30	10,8	170	10	94	8	57	7	2,44	1
133	R-FCN	MobileNet	300	50	15,2	26,5	7,8	1	13,8	30	10,8	159	10	92	8	60	5	2,44	1
134	R-FCN	MobileNet	300	20	14,7	25,9	7,3	1	13,6	28,7	10,8	150	10	96	7	59	5	2,44	1
135	R-FCN	MobileNet	300	10	13,8	24,4	6,5	1	13,1	26,4	10,8	137	11	97	5	58	5	2,44	1
136	R-FCN	MobileNet	600	300	16,5	25,8	13,2	1,5	14	33,8	10,8	607	17	341	25	86	5	9,49	1
137	R-FCN	MobileNet	600	100	17	26,2	13,8	1,8	14,3	35,1	10,8	537	6	296	8	73	6	9,49	1
138	R-FCN	MobileNet	600	50	16,9	26	13,7	2	14,2	34,8	10,8	514	2	283	8	70	6	9,49	1
139	R-FCN	MobileNet	600	20	15,9	24,9	12,6	1,8	13,7	32,3	10,8	511	15	285	8	69	5	9,49	1
140	R-FCN	MobileNet	600	10	14,1	22,2	10,6	1,3	12,5	27,9	10,8	507	20	280	9	67	5	9,49	1

B.3. Bayesian DEA results.

	Architecture	Extractor	Resolution	# Proposals	mAP	mAP (large)	mAP (medium)	mAP (small)	mAP @. 75IOU	mAP @. 50IOU	# model params	Memory	Memory std	CPU	CPU std	GPU	GPU std	FLOPS	DEA result
1	Faster RCNN	VGG	300	300	22,9	37,7	16,4	2,4	22,9	41,1	13,851	1699	81	1286	27	201	16	6432	099,989,176
2	Faster RCNN	VGG	300	10	19,6	32,8	12,7	1,7	20,4	33,4	13,851	1703	30	1322	41	203	16	2912	0,999,729,094
3	Faster RCNN	VGG	300	20	21,5	35,7	14,8	2,1	21,9	37,6	13,851	1870	183	1261	5	204	15	3033	0,993,706,578
4	Faster RCNN	VGG	300	50	22,5	37	16,1	2,3	22,5	40,2	13,851	1842	258	1280	41	202	15	3397	0,973,411,705
5	Faster RCNN	VGG	300	100	22,9	37,5	16,4	2,5	22,8	41	13,851	1713	152	1272	8	201	17	4004	0,997,057,668
6	Faster RCNN	VGG	600	300	25,7	37,2	23	5,6	25,7	46	13,851	2048	210	1668	71	247	14	14,959	0,971,625,067
7	Faster RCNN	VGG	600	10	19	28,3	15,6	3,7	20,1	32,2	13,851	1853	388	1272	7	206	15	11,439	0,970,515,927
8	Faster RCNN	VGG	600	20	22,3	32,8	19,3	5	23,1	38,7	13,851	1984	182	1282	5	205	16	1156	0,987,513,333
9	Faster RCNN	VGG	600	50	24,8	35,6	22,2	5,9	25,3	44	13,851	1822	29	1327	6	209	16	11,925	1
10	Faster RCNN	VGG	600	100	25,6	36,5	22,9	6,2	25,8	45,6	13,851	1948	210	1384	11	217	15	12,532	099,395,933
11	Faster RCNN	Resnet 101	300	300	26,5	43,5	19	3	27,2	44,6	6315	3236	190	2484	28	206	14	23,942	0,713,567,647
12	Faster RCNN	Resnet 101	300	10	23,1	39,3	14,8	2,3	24,6	37,1	6315	560	19	355	13	92	13	2682	1
13	Faster RCNN	Resnet 101	300	20	24,8	41,6	17,1	2,6	26,1	40,9	6315	664	40	427	19	102	15	3415	0,997,407,914
14	Faster RCNN	Resnet 101	300	50	25,8	42,7	18,4	2,9	26,7	43,4	6315	1021	10	634	15	112	17	5614	0,991,107,729
15	Faster RCNN	Resnet 101	300	100	26,3	43,2	18,8	3,1	27	44,3	6315	1459	33	1029	12	136	14	92,8	0,914,321,099
16	Faster RCNN	Resnet 101	600	300	32,2	47,1	28,7	7,5	33,9	52,6	6315	4681	209	3127	17	248	14	29,661	0,952,743,925
17	Faster RCNN	Resnet 101	600	10	26,3	40	21,6	4,6	29,1	40,6	6315	1884	132	967	6	115	15	8402	098,105,357
18	Faster RCNN	Resnet 101	600	20	29,4	44,3	25,3	6,2	31,8	46,4	6315	1985	81	1085	93	126	16	9135	098,744,646
19	Faster RCNN	Resnet 101	600	50	31,3	46,1	27,6	7,2	33,4	50,6	6315	2267	106	1287	46	140	17	11,334	0,989,963,525
20	Faster RCNN	Resnet 101	600	100	32	46,7	28,5	7,5	33,9	52	6315	2756	138	1652	47	164	17	150	0,958,338,596
21	Faster RCNN	Resnet 101	300	300	29,5	46,1	23,4	5,2	31,1	48,3	6315	4271	262	2930	23	233	15	28,999	0,801,307,984
22	Faster RCNN	Resnet 101	300	10	24,5	41,1	16,6	3,5	27	37,9	6315	1378	44	789	23	108	13	7739	0,945,030,885
23	Faster RCNN	Resnet 101	300	20	26,9	43,9	19,9	4,3	29,1	42,7	6315	1464	49	875	10	120	16	8472	0,975,020,567
24	Faster RCNN	Resnet 101	300	50	28,7	45,4	22,3	4,9	30,6	46,5	6315	1703	42	1098	8	132	18	10,671	0,984,282,946
25	Faster RCNN	Resnet 101	300	100	29,2	45,8	23,1	5,2	30,9	47,8	6315	2218	162	1502	12	161	15	14,337	0,916,242,977
26	Faster RCNN	Resnet 101	600	300	34,4	47,7	31,8	10,7	37,1	54,6	6315	8043	301	4859	37	390	15	49,247	0,940,374,452
27	Faster RCNN	Resnet 101	600	10	26,7	40,7	22,2	5,1	29,6	40,2	6315	5056	35	2672	34	196	16	27,987	0,800,751,143
28	Faster RCNN	Resnet 101	600	20	30	44,6	26,8	6,9	33	46,2	6315	5146	143	2729	44	214	15	28,721	0,810,501,842
29	Faster RCNN	Resnet 101	600	50	32,7	46,6	29,9	8,9	35,5	51,2	6315	5375	79	3047	113	234	16	3092	088,662,145
30	Faster RCNN	Resnet 101	600	100	33,7	47,4	31,1	9,9	36,5	53,4	6315	5518	135	3384	65	267	16	34,585	0,917,106,319
31	Faster RCNN	Inception V2	300	300	15,4	27,4	7,7	0,5	14,8	29	1331	2370	51	1388	5	178	15	11,822	0,587,621,848
32	Faster RCNN	Inception V2	300	10	13,3	24,3	5,2	0,3	13,4	23,7	1331	211	6	153	7	79	13	7,63	0,935,649,345
33	Faster RCNN	Inception V2	300	20	14,3	26	6,3	0,3	14,1	26,3	1331	280	11	198	6	90	13	1144	0,847,857,988
34	Faster RCNN	Inception V2	300	50	15	26,9	7,2	0,5	14,5	28	1331	495	21	327	7	99	17	2288	0,719,987,109
35	Faster RCNN	Inception V2	300	100	15,3	27,3	7,4	0,5	14,7	28,5	1331	861	51	535	7	118	15	4195	0,616,208,002
36	Faster RCNN	Inception V2	600	300	21,9	34,4	17,6	3,1	21,8	39,5	1331	2596	63	1611	21	206	15	12,957	0,682,417,576
37	Faster RCNN	Inception V2	600	10	16,4	27,3	10,7	1,9	17,2	28	1331	572	11	383	7	82	12	1898	0,909,673,555
38	Faster RCNN	Inception V2	600	20	19	31,4	13,6	2,2	19,6	33,3	1331	669	16	446	22	98	13	2279	0,897,560,228
39	Faster RCNN	Inception V2	600	50	20,8	33,5	16,1	2,5	21	37,2	1331	871	17	574	22	116	15	3423	0,879,550,227
40	Faster RCNN	Inception V2	600	100	21,5	34	16,9	2,8	21,5	38,7	1331	1152	42	769	16	104	4	53,3	0,901,831,205
41	Faster RCNN	Inception V3	300	300	23,6	40,6	15,1	1,6	23,5	41,4	2627	5016	128	3214	18	353	14	3066	044,984,218
42	Faster RCNN	Inception V3	300	10	20,2	36,1	11,1	1,3	21,3	33,7	2627	337	5	252	9	101	13	1577	0,996,977,915
43	Faster RCNN	Inception V3	300	20	21,9	38,4	12,9	1,6	22,6	37,3	2627	487	13	395	30	109	13	25,8	0,959,931,055
44	Faster RCNN	Inception V3	300	50	23	39,8	14,4	1,7	23,2	40,1	2627	1017	40	669	11	137	13	5589	0,809,622,795
45	Faster RCNN	Inception V3	300	100	23,4	40,3	14,9	1,7	23,4	41	2627	1734	106	1188	22	181	14	10,603	0,638,689,263
46	Faster RCNN	Inception V3	600	300	29,6	44,9	25	5,3	30,6	50,2	2627	5386	143	3662	153	385	13	32,509	0,528,034,572
47	Faster RCNN	Inception V3	600	10	23,9	38,5	17,4	3,2	25,9	38,1	2627	799	71	525	18	116	12	3426	0,960,555,055
48	Faster RCNN	Inception V3	600	20	26,6	41,9	21	4,2	28,3	43,6	2627	1014	21	613	11	130	13	4429	0,994,017,092
49	Faster RCNN	Inception V3	600	50	28,6	44	23,5	4,9	29,9	47,9	2627	1522	14	1048	60	153	16	7437	0,967,415,353
50	Faster RCNN	Inception V3	600	100	29,2	44,6	24,3	5,4	30,2	49,4	2627	2293	81	1482	5	206	13	12,451	090,540,433
51	Faster RCNN	Inception Resnet V2	300	300	28,4	47	20,5	3,6	29,3	47,1	6002	14,769	103	7118	104	625	15	63,998	0,353,609,455
52	Faster RCNN	Inception Resnet V2	300	10	24,7	42,7	15,6	2,8	26,6	38,8	6002	903	14	518	14	130	14	3345	0,982,884,168

(*continued on next page*)

	Architecture	Extractor	Resolution	# Proposals	mAP	mAP (large)	mAP (medium)	mAP (small)	mAP @. 75IOU	mAP @. 50IOU	# model params	Memory	Memory std	CPU	CPU std	GPU	GPU std	FLOPS	DEA result
53	Faster RCNN	Inception Resnet V2	300	20	26,6	45	18,1	3,4	28,1	43	6002	1295	36	756	19	147	13	5436	0,910,115,625
54	Faster RCNN	Inception Resnet V2	300	50	27,9	46,3	19,9	3,7	29	45,9	6002	2622	69	1467	23	197	13	11,711	0,743,147,203
55	Faster RCNN	Inception Resnet V2	300	100	28,2	46,7	20,3	3,7	28,2	46,7	6002	4908	212	2670	16	283	13	22,168	0,594,708,494
56	Faster RCNN	Inception Resnet V2	600	300	35,4	52,8	31,1	8,3	37,5	56,3	6002	15,799	431	7652	31	671	13	68,043	0,694,609,731
57	Faster RCNN	Inception Resnet V2	600	10	29	46,1	22,9	5,4	31,8	43,5	6002	2173	57	1138	28	157	14	73,9	0,934,164,306
58	Faster RCNN	Inception Resnet V2	600	20	32,2	49,6	27,1	6,7	34,9	49,6	6002	2651	50	1297	33	177	14	9481	0,965,421,866
59	Faster RCNN	Inception Resnet V2	600	50	34,2	51,5	29,7	7,8	36,6	54,1	6002	4172	119	2058	62	232	13	15,756	0,858,164,779
60	Faster RCNN	Inception Resnet V2	600	100	34,9	52,4	30,5	8,1	37,1	55,4	6002	6222	257	3390	47	319	14	26,213	0,767,438,361
61	Faster RCNN	Inception Resnet V2	300	300	29,8	48,4	22,8	4,5	30,9	49,2	6002	6053	267	2905	66	272	19	66,545	0,640,190,174
62	Faster RCNN	Inception Resnet V2	300	10	25,1	43,5	16,2	2,7	27,2	39	6002	6207	79	2937	31	287	15	5892	0,577,541,319
63	Faster RCNN	Inception Resnet V2	300	20	27,4	46,2	19,3	3,8	29,4	43,7	6002	6138	109	2917	24	286	16	7984	0,626,636,227
64	Faster RCNN	Inception Resnet V2	300	50	28,9	47,7	21,3	4,4	30,4	47,1	6002	5920	177	3036	15	283	16	14,258	0,642,692,331
65	Faster RCNN	Inception Resnet V2	300	100	29,5	48,1	22,1	4,5	30,7	48,5	6002	6141	81	2945	52	281	17	24,715	061,652,699
66	Faster RCNN	Inception Resnet V2	600	300	35,7	52,5	32	8,9	38	56,5	6002	19,467	284	10,018	491	855	14	78,899	040,432,115
67	Faster RCNN	Inception Resnet V2	600	10	29,1	46,5	23,6	4,9	31,9	43,3	6002	6008	111	2889	34	285	15	18,245	0,636,675,846
68	Faster RCNN	Inception Resnet V2	600	20	32,2	49,8	27,5	6,6	34,9	49,2	6002	6536	186	3230	92	308	13	20,337	0,641,461,964
69	Faster RCNN	Inception Resnet V2	600	50	34,4	51,6	30,4	7,9	36,9	53,8	6002	7854	19	3849	29	367	15	26,611	0,977,820,333
70	Faster RCNN	Inception Resnet V2	600	100	35,3	52,1	31,5	8,5	37,6	55,6	6002	10,341	296	5132	38	465	14	37,069	0,721,956,602
71	Faster RCNN	MobileNet	300	300	16,4	27,8	9,9	1	15,5	31,5	6,06	1147	28	427	22	94	17	2523	0,893,221,145
72	Faster RCNN	MobileNet	300	10	14,4	25,3	7,3	0,8	14,2	26,5	6,06	143	5	100	5	54	7	2,55	099,989,099
73	Faster RCNN	MobileNet	300	20	15,5	27	8,6	0,9	15	29,3	6,06	172	6	123	4	61	9	3,33	0,999,367,307
74	Faster RCNN	MobileNet	300	50	16,2	27,7	9,5	1	15,3	31	6,06	284	4	146	4	72	14	5,67	0,990,888,468
75	Faster RCNN	MobileNet	300	100	16,4	27,9	9,9	0,9	15,5	31,5	6,06	447	17	203	9	81	14	9,59	0,927,673,111
76	Faster RCNN	MobileNet	600	300	19,8	29,3	17,5	3,8	18,3	38,5	6,06	1412	68	610	8	121	16	30,5	0,992,400,523
77	Faster RCNN	MobileNet	600	10	15,1	23,7	11,8	2,4	14,8	28	6,06	487	7	273	12	63	6	7,82	0,983,078,006
78	Faster RCNN	MobileNet	600	20	17,4	26,8	14,4	2,8	16,4	33,2	6,06	507	4	292	2	69	9	8,6	0,999,377,036
79	Faster RCNN	MobileNet	600	50	19,1	28,6	16,7	3,5	17,8	37	6,06	572	25	349	11	79	14	1094	0,999,729,055
80	Faster RCNN	MobileNet	600	100	19,7	29,2	17,4	3,7	18,1	38,3	6,06	768	10	376	18	98	14	1485	0,998,202,182
81	R-FCN	Resnet 101	300	300	25,2	41,1	18,7	2,5	25,9	43,4	6848	992	30	444	15	103	6	2714	1
82	R-FCN	Resnet 101	300	100	25,2	41,2	18,6	2,6	26	43,5	6848	861	47	437	25	94	5	2714	0,998,793,072
83	R-FCN	Resnet 101	300	50	24,9	40,9	1821	2,5	25,8	42,7	6848	798	51	425	20	92	6	2714	0,999,097,611
84	R-FCN	Resnet 101	300	20	23,8	39,7	16,8	2,1	25,2	40,1	6848	713	79	413	10	90	6	2714	0,998,732,677
85	R-FCN	Resnet 101	300	10	22,1	37,6	14,4	1,8	23,9	36,3	6848	708	25	428	31	89	6	2714	0,995,053,005
86	R-FCN	Resnet 101	600	300	30,4	43,3	28,3	7,2	32	51,4	6848	2878	75	1536	42	161	8	1073	0,998,702,076
87	R-FCN	Resnet 101	600	100	30,5	43,4	28,5	7,4	32,1	51,6	6848	2651	19	1465	62	141	9	1073	0,999,719,894
88	R-FCN	Resnet 101	600	50	30,1	43,3	27,9	7,3	31,8	50,4	6848	2591	100	1419	40	134	8	1073	0,995,150,449
89	R-FCN	Resnet 101	600	20	28,3	41,3	25,7	6,2	30,4	46,5	6848	2600	69	1503	103	131	8	1073	0,965,018,477
90	R-FCN	Resnet 101	600	10	25,3	37,9	21,9	4,9	27,7	40,5	6848	2530	34	1438	64	131	8	1073	0,926,073,406
91	R-FCN	Resnet 101	300	300	27,6	43,3	22,7	4,4	29	46,1	6848	2190	26	1330	25	147	7	10,067	0,993,167,256
92	R-FCN	Resnet 101	300	100	27,4	43,5	22,3	4,2	28,9	45,7	6848	2101	97	1314	21	133	6	10,067	0,928,019,446
93	R-FCN	Resnet 101	300	50	27	43,3	21,6	3,7	28,7	44,5	6848	2072	86	1325	57	128	6	10,067	0,897,098,041
94	R-FCN	Resnet 101	300	20	25,2	42	18,6	3	27,4	40,7	6848	2088	86	1303	23	127	6	10,067	0,871,237,262
95	R-FCN	Resnet 101	300	10	23,1	39,4	15,7	2,4	25,5	36,5	6848	1943	93	1326	57	126	6	10,067	0,800,720,967
96	R-FCN	Resnet 101	600	300	31,9	43	30,8	10,4	33,7	52,4	6848	7746	394	4823	145	389	14	39,178	0,772,787,428
97	R-FCN	Resnet 101	600	100	31,7	43,2	30,7	10	33,8	52,2	6848	7143	97	4747	125	327	13	39,178	0,847,051,503
98	R-FCN	Resnet 101	600	50	31,1	42,9	30,1	9,3	33,3	50,8	6848	7842	163	4769	108	313	12	39,178	0,779,429,498
99	R-FCN	Resnet 101	600	20	28,8	41,7	26,9	7,2	31,5	46,1	6848	7562	251	4801	134	304	13	39,178	0,617,586,292
100	R-FCN	Resnet 101	600	10	25,6	38,8	22,6	5,4	28,3	40,1	6848	7553	244	4685	113	302	13	39,178	051,970,279
101	R-FCN	Inception V2	300	300	15,4	27,8	7	0	14,5	29,5	1806	286	14	144	7	72	5	5,09	1
102	R-FCN	Inception V2	300	100	15,5	27,9	7,2	1	14,6	29,9	1806	208	11	125	6	69	6	5,09	0,999,535,091
103	R-FCN	Inception V2	300	50	15,3	27,7	7,1	1	14,5	29,6	1806	202	13	125	6	66	6	5,09	0,998,471,992

(*continued on next page*)

	Architecture	Extractor	Resolution	# Proposals	mAP	mAP (large)	mAP (medium)	mAP (small)	mAP @. 75IOU	mAP @. 50IOU	# model params	Memory	Memory std	CPU	CPU std	GPU	GPU std	FLOPS	DEA result
104	R-FCN	Inception V2	300	20	14,7	26,8	6,5	0	14,1	27,9	1806	177	10	123	8	64	6	5,09	0,999,428,009
105	R-FCN	Inception V2	300	10	13,8	25,2	5,7	0	13,6	25,6	1806	164	12	121	7	66	6	5,09	099,969,596
106	R-FCN	Inception V2	600	300	19,8	31,8	15,2	1,8	18,2	38,3	1806	715	6	445	16	101	6	1985	0,997,671,385
107	R-FCN	Inception V2	600	100	20,1	31,9	15,6	2	18,5	39,1	1806	647	19	394	15	87	6	1985	0,978,431,006
108	R-FCN	Inception V2	600	50	19,7	31,6	15	2	18,3	38,1	1806	624	25	391	19	84	7	1985	0,961,580,826
109	R-FCN	Inception V2	600	20	18,3	30,1	13,2	1,8	17,4	34,7	1806	609	9	378	5	81	6	1985	0,957,027,022
110	R-FCN	Inception V2	600	10	15,9	26,5	10,6	1,6	15,6	29,5	1806	581	12	376	8	79	7	1985	0,890,098,397
111	R-FCN	Inception Resnet V2	300	300	22,5	39,6	13,2	1	22,7	39,8	6506	696	108	339	10	108	6	1499	1
112	R-FCN	Inception Resnet V2	300	100	22,8	39,8	13,8	1,1	23	40,7	6506	642	71	315	13	100	7	1499	0,999,804,391
113	R-FCN	Inception Resnet V2	300	50	22,8	39,6	14	1,2	23	40,6	6506	679	33	314	7	101	6	1499	0,999,906,686
114	R-FCN	Inception Resnet V2	300	20	22,3	38,9	13,5	1	22,8	39,1	6506	611	42	312	13	99	6	1499	0,999,937,914
115	R-FCN	Inception Resnet V2	300	10	21,1	37,5	12,1	1,1	22,1	36	6506	637	96	324	10	98	7	1499	0,997,406,826
116	R-FCN	Inception Resnet V2	600	300	30,7	46,2	26,8	5,6	31,7	52,3	6506	2173	108	1048	24	165	6	6396	0,994,306,459
117	R-FCN	Inception Resnet V2	600	100	30,9	46,4	27	6,1	31,9	52,8	6506	2121	36	993	26	150	6	6396	0,998,005,949
118	R-FCN	Inception Resnet V2	600	50	30,7	46,1	26,8	6,2	32	52,3	6506	2150	83	1060	41	146	6	6396	0,987,753,468
119	R-FCN	Inception Resnet V2	600	20	29,3	44,9	25	5,4	31,1	48,8	6506	2051	12	1007	30	144	6	6396	0,996,588,879
120	R-FCN	Inception Resnet V2	600	10	27,1	42,7	22,1	4,4	29,3	43,9	6506	2151	62	1034	12	141	5	6396	0,949,050,948
121	R-FCN	Inception Resnet V2	300	300	26	43,5	18,7	2	26,4	45,3	6506	1609	37	827	14	142	7	4774	0,964,248,588
122	R-FCN	Inception Resnet V2	300	100	26	43,5	18,6	2,1	26,4	45,2	6506	1557	28	841	10	134	6	4774	0,943,934,531
123	R-FCN	Inception Resnet V2	300	50	25,6	43,2	17,9	1,9	26,1	44,3	6506	1564	14	808	5	133	7	4774	0,973,978,512
124	R-FCN	Inception Resnet V2	300	20	24,4	42,4	15,9	1,7	25,4	41,3	6506	1499	83	796	3	128	8	4774	0,892,191,711
125	R-FCN	Inception Resnet V2	300	10	22,5	40,1	13,5	1,5	23,9	37,4	6506	1501	102	791	14	131	6	4774	0,805,923,613
126	R-FCN	Inception Resnet V2	600	300	32,3	46,9	29,3	8	33,8	53,9	6506	7289	46	3220	60	368	8	20,545	094,355,266
127	R-FCN	Inception Resnet V2	600	100	32,1	46,8	29,3	8	33,7	53,7	6506	6653	53	3319	70	328	6	20,545	0,850,951,265
128	R-FCN	Inception Resnet V2	600	50	31,5	46,4	28,6	7,6	33,3	52,2	6506	6771	258	3315	32	313	7	20,545	0,805,416,577
129	R-FCN	Inception Resnet V2	600	20	29,7	45,2	26,4	6,7	32	48,3	6506	6958	137	3318	100	302	7	20,545	0,642,626,419
130	R-FCN	Inception Resnet V2	600	10	27,1	42,7	22,7	4,8	29,7	42,9	6506	7001	40	3320	21	299	7	20,545	0,728,464,914
131	R-FCN	MobileNet	300	300	15	26,3	7,6	0	13,7	29,4	10,8	246	4	114	4	66	5	2,44	1
132	R-FCN	MobileNet	300	100	15,2	26,5	7,9	1	13,8	30	10,8	170	10	94	8	57	7	2,44	0,999,939,609
133	R-FCN	MobileNet	300	50	15,2	26,5	7,8	1	13,8	30	10,8	159	10	92	8	60	5	2,44	099,971,832
134	R-FCN	MobileNet	300	20	14,7	25,9	7,3	1	13,6	28,7	10,8	150	10	96	7	59	5	2,44	0,999,863,575
135	R-FCN	MobileNet	300	10	13,8	24,4	6,5	1	13,1	26,4	10,8	137	11	97	5	58	5	2,44	0,999,592,874
136	R-FCN	MobileNet	600	300	16,5	25,8	13,2	1,5	14	33,8	10,8	607	17	341	25	86	5	9,49	0,996,511,331
137	R-FCN	MobileNet	600	100	17	26,2	13,8	1,8	14,3	35,1	10,8	537	6	296	8	73	6	9,49	0,992,550,738
138	R-FCN	MobileNet	600	50	16,9	26	13,7	2	14,2	34,8	10,8	514	2	283	8	70	6	9,49	0,994,425,298
139	R-FCN	MobileNet	600	20	15,9	24,9	12,6	1,8	13,7	32,3	10,8	511	15	285	8	69	5	9,49	0,969,886,745
140	R-FCN	MobileNet	600	10	14,1	22,2	10,6	1,3	12,5	27,9	10,8	507	20	280	9	67	5	9,49	0,920353119

References

[1] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, et al., Speed/accuracy trade-offs for modern convolutional object detectors, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 7310–7311.

[2] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, F.E. Alsaadi, A survey of deep neural network architectures and their applications, Neurocomputing 234 (2017) 11–26, doi:10.1016/j.neucom.2016.12.038.

[3] D. Wu, S.-J. Zheng, X.-P. Zhang, C.-A. Yuan, F. Cheng, Y. Zhao, Y.-J. Lin, Z.-Q. Zhao, Y.-L. Jiang, D.-S. Huang, Deep learning-based methods for person re-identification: a comprehensive review, Neurocomputing 337 (2019) 354–371, doi:10.1016/j.neucom.2019.01.079.

[4] Y. Yuan, G. Xun, Q. Suo, K. Jia, A. Zhang, Wave2vec: deep representation learning for clinical temporal data, Neurocomputing 324 (2019) 31–42. Deep Learning for Biological/Clinical Data, doi: 10.1016/j.neucom.2018.03.074.

[5] H. Cuayuitl, A data-efficient deep learning approach for deployable multimodal social robots, Neurocomputing (2019), doi:10.1016/j.neucom.2018.09.104.

[6] X.W. Gao, C. James-Reynolds, E. Currie, Analysis of tuberculosis severity levels from ct pulmonary images based on enhanced residual deep learning architecture, Neurocomputing (2019), doi:10.1016/j.neucom.2018.12.086.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009, IEEE, 2009, pp. 248–255.

[8] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, A. Zisserman, The pascal visual object classes (VOC) challenge, Int. J. Comput. Vis. 88 (2) (2010) 303–338, doi:10.1007/s11263-009-0275-4.

[9] T.Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft COCO: common objects in context, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), in: LNCS, 8693, 2014, pp. 740–755, doi:10.1007/978-3-319-10602-1_48.

[10] A. Charnes, W.W. Cooper, E. Rhodes, Measuring the efficiency of decision making units, Eur. J. Oper. Res. 2 (6) (1978) 429–444.

[11] A. Fernandez-Montes, F. Velasco, J. Ortega, Evaluating decision-making performance in a grid-computing environment using dea, Expert Syst. Appl. 39 (15) (2012) 12061–12070.

[12] D. Fernández-Cerero, A. Fernández-Montes, F. Velasco, Productive efficiency of energy-aware data centers, Energies 11 (8) (2018), doi:10.3390/en11082053.

[13] A. Amirteimoori, A. Emrouznejad, Optimal input/output reduction in production processes, Decis Support Syst. 52 (3) (2012) 742–747.

[14] C. Kao, S.-N. Hwang, Efficiency measurement for network systems: it impact on firm performance, Decis Support Syst. 48 (3) (2010) 437–446.

[15] H. Eilat, B. Golany, A. Shtub, R&d project evaluation: an integrated dea and balanced scorecard approach, Omega (Westport) 36 (5) (2008) 895–912.

[16] A. Emrouznejad, B.R. Parker, G. Tavares, Evaluation of research in efficiency and productivity: a survey and analysis of the first 30 years of scholarly literature in dea, Socioecon. Plann. Sci. 42 (3) (2008) 151–157.

[17] A. Expósito-García, F. Velasco-Morente, How efficient are universities at publishing research? a data envelopment analysis of spanish state universities, El profesional de la información 27 (5) (2018) 1108–1115.

[18] A. Afonso, L. Schuknecht, V. Tanzi, Public sector efficiency: evidence for new eu member states and emerging markets, Appl. Econ. 42 (17) (2010) 2147–2164.

[19] M.R. González-Rodríguez, F. Velasco-Morente, L. González-Abril, La eficiencia del sistema de protección social español en la reducción de la pobreza, Papeles de población 16 (64) (2010) 123–154.

[20] M. Campos, A. Fernandez-Montes, J. Gavilan, F. Velasco, Public resource usage in health systems: a data envelopment analysis of the efficiency of health systems of autonomous communities in spain, Public Health 138 (2016) 33–40.

[21] J. Fernández-Serrano, V. Berbegal, F. Velasco, A. Expósito, Efficient entrepreneurial culture: a cross-country analysis of developed countries, Int. Entrepr. Manag. J. 14 (1) (2018) 105–127.

[22] A. Exposito, F. Velasco, Municipal solid-waste recycling market and the european 2020 horizon strategy: a regional efficiency analysis in spain, J. Clean Prod. 172 (2018) 938–948.

[23] M.T. Sanz-Díaz, F. Velasco-Morente, R. Yñiguez, E. Díaz-Calleja, An analysis of spain's global and environmental efficiency from a european union perspective, Energy Policy 104 (2017) 183–193.

[24] J.-J. Moreno-Moreno, F.V. Morente, M.T.S. Díaz, Assessment of the operational and environmental efficiency of agriculture in latin america and the caribbean, Agricult. Econ. 64 (2) (2018) 74–88.

[25] A. Expósito, J. Fernández-Serrano, F. Velasco, Crecimiento económico, pobreza y desigualdad: un análisis de eficiencia para américa latina en el siglo xxi, Revista de Economía Mundial (47) (2017) 117–138.

[26] J.-J. Moreno-Moreno, T. Sanz-Diaz, F. Velasco-Morente, C. Ludena, A dea-based evaluation of latin america and the caribbean agricultural environmental performance under the assumption of natural and managerial efficiency, Revista de Economía Mundial (47) (2017) 157–178.

[27] A. Charnes, W.W. Cooper, E. Rhodes, Evaluating program and managerial efficiency: an application of data envelopment analysis to program follow through, Manage Sci. 27 (6) (1981) 668–697.

[28] R.D. Banker, A. Charnes, W.W. Cooper, Some models for estimating technical and scale inefficiencies in data envelopment analysis, Manage Sci 30 (9) (1984) 1078–1092.

[29] R. Barkhi, Y.-C. Kao, Evaluating decision making performance in the GDSS environment using data envelopment analysis, Decis. Support Syst. 49 (2) (2010) 162–174.

[30] R. Ranjan, V.M. Patel, R. Chellappa, Hyperface: a deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition, IEEE Trans. Pattern Anal. Mach. Intell. 41 (1) (2019) 121–135.

[31] R. Olmos, S. Tabik, F. Herrera, Automatic handgun detection alarm in videos using deep learning, Neurocomputing 275 (2018) 66–72.

[32] Á. Arcos-García, J.A. Álvarez-García, L.M. Soria-Morillo, Deep neural network for traffic sign recognition systems: an analysis of spatial transformers and stochastic optimisation methods, Neural Netw. 99 (2018) 158–165.

[33] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, in: Proceedings of the Advances in Neural Information Processing Systems, 2015, pp. 91–99.

[34] J. Dai, Y. Li, K. He, J. Sun, R-fcn: Object detection via region-based fully convolutional networks, in: Proceedings of the Advances in Neural Information Processing Systems, 2016, pp. 379–387.

[35] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.Y. Fu, A.C. Berg, SSD: Single shot multibox detector, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), in: LNCS, 9905, 2016, pp. 21–37, doi:10.1007/978-3-319-46448-0_2.

[36] J. Redmon, A. Farhadi, Yolo9000: Better, faster, stronger, in: Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6517–6525, doi:10.1109/CVPR.2017.690.

[37] A. Arcos-Garcia, J.A. Álvarez-García, L.M. Soria-Morillo, Evaluation of deep neural networks for traffic sign detection systems, Neurocomputing 316 (2018) 332–344.

[38] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv:1409.1556 (2014).

[39] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[40] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, arXiv:1502.03167 (2015).

[41] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: Proceedings of the IEEE conference on Computer Vision and pattern Recognition, 2016, pp. 2818–2826.

[42] C. Szegedy, S. Ioffe, V. Vanhoucke, A.A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning., in: Proceeding of the AAAI, 2017, pp. 4278–4284.

[43] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: efficient convolutional neural networks for mobile vision applications, arXiv:1704.04861 (2017).

[44] O.B. Olesen, N.C. Petersen, Stochastic data envelopment analysis a review, Eur. J. Oper. Res. 251 (1) (2016) 2–21.

[45] A. Charnes, W.W. Cooper, Chance-constrained programming, Manage Sci. 6 (1) (1959) 73–79.

[46] K.C. Land, C.K. Lovell, S. Thore, Chance-constrained data envelopment analysis, Manag. Decision Econ. 14 (6) (1993) 541–554.

[47] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2117–2125.

[48] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2961–2969.

[49] M. Najibi, M. Rastegari, L.S. Davis, G-cnn: an iterative grid based object detector, in: Proceedings of the IEEE Conference on Computer Vision and Pattern recognition, 2016, pp. 2369–2377.

[50] T. Sueyoshi, M. Goto, Dea radial measurement for environmental assessment: a comparative study between japanese chemical and pharmaceutical firms, Appl. Energy 115 (2014) 502–513.

[51] Z. Huang, S.X. Li, Stochastic dea models with different types of input–output disturbances, J. Product. Anal. 15 (2) (2001) 95–113.

[52] A. Charnes, W.W. Cooper, Management models and industrial applications of linear programming, Manage. Sci. 4 (1) (1957) 38–91.

[53] A.E. Gelfand, A.F. Smith, Sampling-based approaches to calculating marginal densities, J. Am. Stat. Assoc. 85 (410) (1990) 398–409.

[54] M.-H. Chen, Q.-M. Shao, J.G. Ibrahim, Monte Carlo Methods in Bayesian Computation, Springer Science & Business Media, 2012.

[55] F.J. Ortega, J.M. Gavilan, Bayesian estimation of the half-normal regression model with deterministic frontier, Comput. Stat. 31 (3) (2016) 1059–1078.

[56] G.E. Box, G.C. Tiao, Bayesian Inference in Statistical Analysis, Addison-Wesley, 1973. Reading, Massachussetts

[57] F.J. Ortega, J. Basulto, A generalization of Jeffreys' rule for non regular models, Commun. Stat. Theory Methods 45 (15) (2016) 4433–4444.