Towards an Image Utility Assessment Framework for Machine Perception

Zohaib Amjad Khan, Aladine Chetouani, Giuseppe Valenzise, Frédéric Dufaux

Towards an Image Utility Assessment Framework for Machine Perception

Zohaib Amjad Khan[1], Giuseppe Valenzise[1], Aladine Chetouani[2], Frédéric Dufaux[1]
[1]*Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des signaux et systèmes, Gif-sur-Yvette, France*
[2]*Laboratoire PRISME, Université d'Orléans, Orléans, France,*

[1](zohaib.khan, giuseppe.valenzise, frederic.dufaux)@l2s.centralesupelec.fr
[2]aladine.chetouani@univ-orleans.fr

***Abstract*—In real-world applications, images and videos used in computer vision algorithms are often distorted due, e.g., to compression and transmission. As a result, they may lose relevant information content, or they may deviate significantly from the original data distribution used to train the machine task, rendering the visual content practically useless with respect to its initial purpose. Evaluating the *utility* of an image for machine tasks has received little attention so far in the literature. This concept of utility is substantially different from the visual quality typically used in image/video compression, as the latter is related to the perception of the human visual system. In this paper, we propose a definition of utility as the degree of confidence by which a machine task is able to take a decision. In this context, we propose a *full-reference* utility loss measure: we assume that the decision on the pristine image is correct (reference), and we measure the utility loss as the confidence reduction in the decision due to a noisy input with respect to this reference. We apply this general definition on two specific tasks, classification and object detection, and we study practical solutions to predict utility, as well as the ability of our utility measure to generalize across tasks.**

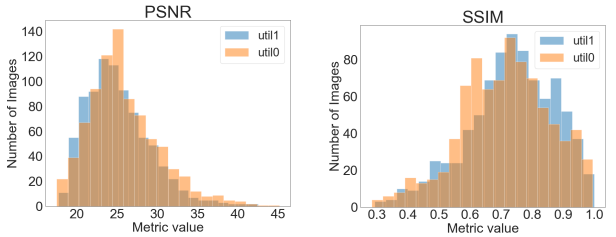
***Index Terms*—image utility for machines, machine perception, task-based assessment, image utility assessment**

I. Introduction

It is well known that the concept of visual quality for the human visual system (HVS) is different from the quality of an input image/video to perform a computer vision task [1] [2]. For example, a machine-based task may produce the same outcomes for two images with big differences in visual image quality and vice versa. In real world scenarios, the distortion introduced during acquisition and transmission impacts the utility of images for the machine tasks, which are generally trained on pristine examples [3]. Retraining the networks with distorted images in this case is not only time-consuming and unfeasible, but also an intrusive and a non-generalizable solution. Instead, having a measure of utility could be used as an optimization constraint by machine tasks [4] or as a more generalized and non-intrusive solution for improving the image utility and machine performance.

Existing image quality assessment methods have been designed by modeling human perception, and are typically trained on datasets with subjective mean opinion scores labels.

This work was funded by BPI France in the framework of the SMART-V2I project.



(a) PSNR (b) SSIM [5]

(c) OOD [6] (d) Adversarial [7]

Fig. 1: Histograms for different metrics with binary utility

Image quality in this context is defined in terms of either the level of visual impairment in the image, or its perceptual quality compared to a pristine reference [5]. This makes visual quality metrics essentially unsuitable to predict image utility for machines. Moreover, it is pertinent to mention here that the concept of machine image utility assessment is not to be confused with anomaly detection often used in relation to Deep Learning (DL) methods. Anomaly detection methods such as out-of-distribution (OOD) detection and adversarial image detection are concerned with special cases only, whereas the concept of image utility for machines is universal and applies to normal application scenarios as given by examples before. OOD images belong to the class of images that are different to the training set, whereas adversarial images are similar to training images but have intentional non-perceptual perturbations to create mis-classification errors. None of these two scenarios applies here. Fig. 1 illustrates how these other notions are unsuitable for assessing image utility with a simple example. For this example, a small set of images from imagenet [8] validation data were selected, from which another set was generated by compressing them at different bitrates. Both these compressed and uncompressed images were then

passed through a pretrained DL-based classification network. Each plot in the figure consists of two histograms, where the blue histogram represents those compressed images that are given the same labels as their uncompressed versions and the orange histogram shows the frequency of wrongly classified images. Each plot shows a representative method from among the quality assessment and anomaly detection class of methods [5]–[7]. It can be seen from the plots that none of the existing notions capture the concept of image utility for machines as there is a large overlap between the two histograms in each plot.

Therefore, in order to assess the image utility there is a need to first formally define this concept and then based on that to design an effective methodology for evaluating it. This work is an initial contribution in this regard as we present the notion of image utility for machines and a methodology for its evaluation followed by exploring its use and effectiveness.

II. Related Work

Recently, some attention has been given in understanding the concept of machine perception and its differences from human perception. Generally, this is being done either by understanding how an input image affects the different parts of the network or by studying the properties of images which are suitable for the machines. For instance, in [9], the authors have proposed to find the convolutional filters most affected by the distortions introduced during image acquisition and transmission, and then have proposed a correcting mechanism for those. On the other hand, in [3], the authors have studied the differences between human and Deep Neural Network (DNN) classification performance for images distorted by blur and noise. They have shown that humans outperform machine based classifiers even when they are retrained using distorted images. Besides these, some other works like [10], [11] have also proposed image correction and recovery methods to improve the outcomes of DL networks in case of distorted images without retraining. Another interesting work [12] has recently explored the concept of Just Noticeable Difference (JND) for machine vision, whereby they have shown that like for HVS, it is possible to find JND for machines also. However, all of these works focus on single machine task i.e. image classification and do not focus on evaluating the overall image utility.

With regards to machine utility assessment, the most relevant work has been done in [13] for videos, where authors have named it as analytical quality and have only considered the classification task. In that work, the authors have proposed to use a mean classifier opinion score (MCOS) instead of Mean Opinion Score (MOS) for training quality assessment networks. MCOS is evaluated using average of a combination of the class confidence and class rank from 5 different classifiers. Then, they have used different thresholds in different applications to filter out frames. However, in that work they only focus on image classification and then use the same MCOS for all other machine tasks. Contrary to that, in this work we will first consider image utility for machines in general, before providing a task-specific definition as done in the next section.

III. Image Utility for Machines

For a specific machine task like image classification, face recognition, object detection etc., we can define image utility in terms of the confidence in prediction (no-reference) or change in prediction confidence from that of a pristine image prediction (full-reference). Based on that, we first define here image utility for machines in general terms below, followed by task-specific definition.

A. General Definition

Let's consider a pristine image I' and a distorted image I both of which have been passed through a DNN i for a specific task. Let's also consider $\mathcal{S}$ as the set of M possible outcomes for the task, which also contains the ground-truth for images I' and I i.e. $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, ..\mathcal{S}_M\}$. When an image is passed through i, the output is a prediction $\mathcal{P}$ such that $\mathcal{P}_i(I) \in \mathcal{S}$ along with a set of confidence values associated for each member of the outcomes set. If ϕ_i' is the confidence value assigned to the ground-truth when I' is passed and ϕ_i is the confidence value assigned to the ground-truth when I is passed, then utility $\mathcal{U}_i$ for the DNN i can be defined as a function of ϕ_i' and ϕ_i as

$$\mathcal{U}_i(I) = f(\phi_i, \phi_i') \tag{1}$$

If ϕ_i^{rec} is the recognition confidence threshold below which the prediction is rejected, we propose to define $f(\phi_i, \phi_i')$ as

$$f(\phi_i, \phi_i') = \frac{\phi_i - \phi_i^{rec}}{1 - (\phi_i' - \phi_i^{rec})} \tag{2}$$

The higher the value of this function, the higher the utility value. For the case where the distorted image gives the wrong prediction i.e. $\phi_i < \phi_i^{rec}$, this becomes negative. If we have a set of possible DNNs for a specific task, then image utility for machines $\mathcal{U}$ can be defined as

$$\mathcal{U}(I) = \mathcal{O}(U) \tag{3}$$

where $\mathcal{O}$ is some kind of operator like mean, max or majority vote and U is a set of utility values of N different DNNs i.e.

$$U(I) = \{\mathcal{U}_1(I), \mathcal{U}_2(I), .., \mathcal{U}_N(I)\} \tag{4}$$

B. Task-Specific Utility

It is possible to extend and simplify the general Eqs.(1-3) to any specific machine task, $\mathcal{T}$. In the following, we show how this notion of image utility may be simplified for the two most common machine vision tasks of image classification and object detection.

1) Image Classification: Let's consider the simplest example of image classification first with N classifiers and consider all images with wrong predicted class or $f(\phi_i, \phi_i') < 0$ as having no utility and those with positive values as 'useful'. Hence, the utility function for each individual network with $\mathcal{T} = \mathcal{C}$ becomes

$$\mathcal{U}_i^{(\mathcal{C})}(I) = \begin{cases} 1, & \text{if } \mathcal{P}_i(I) = \mathcal{S}(I) \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where $\mathcal{P}_i(I)$ and $\mathcal{S}(I)$ are the predicted and the ground-truth classes respectively. The combined utility from all the networks in this case $\mathcal{U}^{(\mathcal{C})}(I)$ can then be evaluated using some kind of operator like majority voting.

2) Object Detection: Let's consider another task of object detection where $\mathcal{T} = \mathcal{OD}$. The utility of an object detector lies in two aspects namely correct prediction of objects and their correct localization. Hence, each outcome in the set $\mathcal{S}$ in this case is a subset consisting of a label L and the localization box B i.e. $\mathcal{S}_j = \{L_j, B_j\}$. The ground-truth for an image may be one or multiple of these outcome subsets. For L, we apply the same condition of $f(\phi_i, \phi_i') < 0$ for no utility and vice versa. For correct localization we can use a threshold of 0.5 on its associated confidence score ϕ_{loc}. Let's consider the simplest case where ground-truth consists of a single possible outcome. Hence, individual detector utility in this case becomes

$$\mathcal{U}_i^{(\mathcal{OD})}(I) = \begin{cases} 1, & \text{if } \mathcal{P}_i(I) = \mathcal{S}(I) \text{ and } \phi_{loc} > 0.5 \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

The combined image utility for object detection task can then be obtained the same way as for classification.

It is not feasible or in any way useful to evaluate these utility functions by passing the image through the inputting network of the task. The real usage of this comes if we know the utility of the image beforehand. Hence, there is a need to predict value of this utility function for each input image by having a utility assessment module. In the next section, we propose one such assessment method.

IV. Machine Image Utility Assessment

For our proposed machine image utility assessment method, we consider a two-class formulation consisting of 'useful' and 'non-useful' labels as defined by the utility functions $\mathcal{U}^{(\mathcal{C})}$ and $\mathcal{U}^{(\mathcal{OD})}$ resulting from combining outputs from Eq.(5) and (6). These labels are generated using a standard DNN like Inception-v3 [14] for classification and a standard object detector like Faster R-CNN [15]. For classification, we also include here a threshold on the correct class softmax probability (CCSP) in the definition. Hence, the 'useful' label is given to an image whose predicted class is the same as the original image with a CCSP exceeding a threshold T_H. On the other hand, any image which is wrongly classified by the DNN with a CCSP below a threshold T_L is given a label 'non-useful'. Similarly, for object detection, we add an additional constraint of the Intersection over Union (IoU) value threshold. If at least one of the boxes in the image is detected with an IoU value of greater than and equal to 0.5, we consider it as an image with utility 1. On the other hand, all those images for which none of the boxes are detected with an IoU value of greater than or equal to 0.5 are considered as images with utility 0.

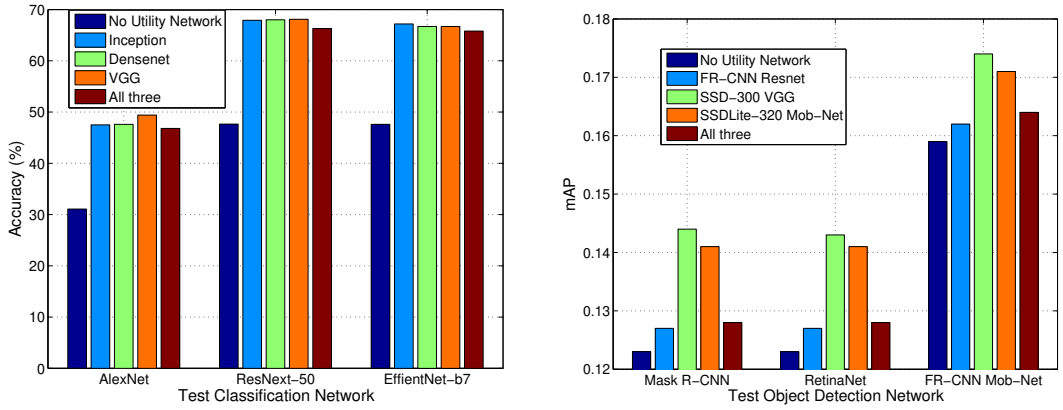
To understand the labeling process, let us consider an image I' that has an associated ground-truth $\mathcal{S}(I')$. In this work, we consider the specific case of JPEG image compression. In order to label the images, we first obtain the label of the uncompressed image $\mathcal{P}(I')$ using the labeling network. Then we compress the image using JPEG compression to get I and pass it through the same network to get its label $\mathcal{P}(I)$. For the classification task, if $\mathcal{P}(I')$ and $\mathcal{P}(I)$ are the same and also match the ground-truth label $\mathcal{S}(I')$ with CCSP exceeding T_H, I is given a utility label of 1 where utility label belongs to one of the two classes in set $\{0, 1\}$. Here class 0 represents a 'non-useful' image and class 1 represents a 'useful' image. On the other hand, if $\mathcal{P}(I') = \mathcal{S}(I')$ but $\mathcal{P}(I)$ is different with CCSP less than T_L, I is given a utility label of 0. This way, we consider only the impact of distortion on utility. Similarly, for the object detection if at least one of the many predicted outcomes $\mathcal{P}(I')$ and $\mathcal{P}(I)$ are the same and also match the ground-truth label $\mathcal{S}(I')$ with confidence value and IoU both greater or equal to 0.5, it is given the utility value of 1. All other images with no right labels or having lower confidence scores or IoU values are given a utility label of 0. Finally, for combining the labels from multiple networks, the images assigned utility 1 from all networks are given the label 1, whereas all those with 0 assigned from all networks are considered as utility 0.

Once the labels are generated, we train a binary utility assessment network using these compressed images. In this work, we propose to use a baseline Resnet-18 network [16] for utility assessment. The loss function used during training is the cross-entropy loss. The trained network can then be used to identify the useful and non-useful images from test images.

V. Experimental Results

Besides finding out the accuracy of our assessment network for different tasks, we have evaluated our proposed utility assessment framework with focus on two questions (i) Can we use the utility evaluated for one network for all others in a single task? and (ii) Can we use the utility evaluated for one machine task for another (inter-task generalization)? We have considered two machine tasks here namely image classification and object detection.

For classification, we have used the Imagenet validation set [8] (50000 images) to generate the JPEG compressed images at 9 different compression ratios. For classification networks, we have used three standard networks namely the Inception-v3 [14], VGG-19 [17] and Densenet-121 [18]. For CCSP, the two thresholds T_H and T_L used were 0.9 and 0.1 respectively. Once the labels were generated, we divided the dataset into training and test set with a 80-20 ratio (around 200,000 training images and 50000 test images per class) ensuring that the images in the two sets do not overlap. In order to balance the images for the two utility classes, we also added images for



(a) (b)

Fig. 2: Impact of using the proposed utility assessment method trained for utility of (a) different classification networks on accuracy of test classification networks and of (b) different object detection networks on mAP of test detection networks

TABLE I: Performance of utility assessment network trained for utility of different *classification* networks

Classification Network for defining utility	Accuracy
Inception-v3	87.5%
VGG-19	84.2%
Densenet-121	91.2%
Inception+VGG+Densenet	93.3%

TABLE II: Performance of utility assessment network trained for utility of different *object detection* networks

Object Detection Network for defining utility	Accuracy
Faster R-CNN Resnet-50	84.8%
SSD-300 VGG-16	81.3%
SSD-320 Mobilenet-v3	83.1%
SSD-300 + SSD-320 + Faster R-CNN	92.8%

utility 0 class using data augmentation techniques followed by relabeling. Finally the Resnet-18 was trained using Stochastic Gradient Descent method with a learning rate of 0.0001.

Similarly, for object detection we have taken the COCO 2017 validation set [19] (5000 images) and generated the JPEG compressed images at 9 different compression ratios from them. For carrying out object detection, we have selected three common models namely Faster-RCNN [15], SSD-300 [20] and SSDLite-320 [21], [22]. The train-test ratio used was again 80-20 (around 20000 training images and 5000 test images per class) and images for utility 0 class added using data augmentation for class balancing. The same network, Resnet-18 was then trained for utility prediction using the same hyperparameters.

A. Results from individual and combined networks

Table I shows the accuracy performance of our proposed utility assessment network for different classification networks. We can see that there are slight differences in performance of the network when the classification network is changed for defining utility. For individual networks, Densenet-121 gives the best accuracy results of 91.2% followed by Inception with 87.5% and VGG-19 with 84.2%. We have also evaluated the results for a combined utility evaluated from all these networks using common predicted class. This resulted in a better training than the individual ones as we obtained the binary utility classification accuracy of 93.3%.

Similarly, Table II lists the accuracy of the utility assessment network for different object detectors. Here again, we can see that the combined utility defined as the common output from different object detectors gives the best classification accuracy of 92.8%. For prediction of utility for individual detectors, the classification accuracy is much worse in this case with the second best accuracy of only around 84.8% by the Faster R-CNN Resnet-50 method. The proposed method gives the worst performance for SSD-300 VGG-16 detector with classification accuracy of only 81.8%.

B. Intra-task generalization across machine types (networks)

One of the fundamental benefits of the utility assessment module is to be able to deploy it before any network to identify the useful inputs only and filter out the rest or enhance them before reuse. Ideally, any utility assessment method should be independent of the classification network and work well for all kinds of applications. To evaluate this generalization ability of our utility assessment network for the classification task, we train it for one classification network or a combination of them and then observe the change in performance for other networks after filtering out the images having 0 utility. These other networks selected are AlexNet [23], ResNext-50 [24] and EfficientNet-b7 [25]. Similarly, for evaluating the generalization for the object detection task, we train it for one object detector or a combination of them and then observe the change in performance for other detectors after replacing the images with 0 utility with the uncompressed ones to allow for mAP evaluation. The other object detectors used for this experiment are Mask R-CNN [26], RetinaNet [27] and Faster R-CNN MobileNet-v3 [15], [22]. We used subsets of imagenet and COCO validation datasets with images compressed at bitrates other than used for training for classification and object detection respectively. Figure 2(a) shows how using the utility network impacts the performance of other classification networks with a clear improvement in accuracy in all the tested networks. This implies that our proposed network generalizes significantly well across different networks, although there is still a big room of improvement to get the ideal results. Similarly, Figure 2(b) illustrates the impact of using utility

TABLE III: Training Impact for Classification Accuracies

Network	Classification trained	OD trained
AlexNet	46.8%	46.0%
ResNext-50	66.3%	65.4%
EfficientNet-B7	65.8%	65.9%

TABLE IV: Training Impact for Object Detection mAPs

Network	OD trained	Classification trained
Mask R-CNN	0.128	0.126
RetinaNet	0.128	0.125
FR-CNN MobileNet	0.164	0.161

network for the object detectors. Here, we can see that mean Average Precision (mAP) values (though low due to compression artifacts) increase when the utility network is used although this increase is smaller when utility assessment network is trained for multiple object detectors utility.

C. Inter-task generalization across machine tasks

In order to evaluate whether the utility prediction network can be used globally when trained on one single task, we have also evaluated the inter-task generalization. To do this, we used the network trained for classification utility (from 3 combined networks) and evaluated its impact on object detection task and vice versa. Tables III and IV show the difference in accuracy and mAP values for classification and object detection tasks respectively. From the tables, we can clearly observe that only negligible difference in values (around 2% or less) is observed when utility training from one task is used to detect utility for another task. Hence, this shows that our proposed framework is generalizable across tasks.

VI. Conclusion

In this work, we have presented a formal specification and definition of a very important concept of image utility for machines for the very first time. We have further presented a simple strategy for predicting this machine image utility using training with outputs of multiple classification networks as well as of multiple object detectors. Experimental results have shown the effectiveness and generalization of our proposed method. Hence, such a method of predicting image utility for machines can not only be used in real-world applications to improve outcomes where images are affected by common distortions, but also for optimizing a machine task output for another subsequent machine task. This work can serve as an important first step in building more efficient machine utility assessment methods and for developing algorithms and networks optimized for machine perception. For future work, we also plan to extend this work to include more distortions.

References

[1] J. Borowski, C. M. Funke, K. Stosio, W. Brendel, T. Wallis, and M. Bethge, "The notorious difficulty of comparing human and machine perception," in *2019 Conference on Cognitive Computational Neuroscience*, 2019, pp. 2019–1295.

[2] S. Ling, P. L. Callet, and Z. Yu, "The role of structure and textural information in image utility and quality assessment tasks," *Electronic Imaging*, vol. 2018, no. 14, pp. 1–13, 2018.

[3] S. Dodge and L. Karam, "Human and dnn classification performance on images with quality distortions: A comparative study," *ACM Transactions on Applied Perception (TAP)*, vol. 16, no. 2, pp. 1–17, 2019.

[4] F. Codevilla, J. G. Simard, R. Goroshin, and C. Pal, "Learned image compression for machine perception," *arXiv preprint arXiv:2111.02249*, 2021.

[5] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[6] W. Liu, X. Wang, J. Owens, and Y. Li, "Energy-based out-of-distribution detection," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 464–21 475, 2020.

[7] J. Martin and C. Elster, "Inspecting adversarial examples using the fisher information," *Neurocomputing*, vol. 382, pp. 80–86, 2020.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[9] T. S. Borkar and L. J. Karam, "Deepcorrect: Correcting dnn models against image distortions," *IEEE Transactions on Image Processing*, vol. 28, no. 12, pp. 6022–6034, 2019.

[10] X. Lin, D. Bhattacharjee, M. El Helou, and S. Süsstrunk, "Fidelity estimation improves noisy-image classification with pretrained networks," *IEEE Signal Processing Letters*, vol. 28, pp. 1719–1723, 2021.

[11] M. Takagi, A. Sakurai, and M. Hagiwara, "Quality recovery for image recognition," *IEEE Access*, vol. 7, pp. 105 851–105 862, 2019.

[12] J. Jin, X. Zhang, X. Fu, H. Zhang, W. Lin, J. Lou, and Y. Zhao, "Just noticeable difference for deep machine vision," *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.

[13] S. Paul, U. Drolia, Y. C. Hu, and S. T. Chakradhar, "Aqua: Analytical quality assessment for optimizing video analytics systems," *arXiv preprint arXiv:2101.09752*, 2021.

[14] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and 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.

[15] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[17] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[18] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[19] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[20] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[21] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[22] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan *et al.*, "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1314–1324.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[24] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492–1500.

[25] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.

[26] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[27] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.