Trustworthy AI in Video Surveillance: The IMMAGINA Project

Emanuele Ledda[1,2,*], Lorenzo Putzu[3], Rita Delussu[3], Giorgio Fumera[3] and Fabio Roli[2]

[1]Department of Computer, Control and Management Engineering, Sapienza University of Rome, Via Ariosto 25, Rome, 00185, Italy
[2]Department of Informatics, Bioengineering, Robotics, and Systems Engineering, University of Genova, Via Dodecaneso 35, Genova, 16146, Italy
[3]Department of Electric and Electronic Engineering, University of Cagliari, Via Marengo 3, Cagliari, 09100, Italy

Abstract

The increasing adoption of machine learning and deep learning models in critical applications raises the issue of ensuring their trustworthiness, which can be addressed by quantifying the *uncertainty* of their predictions. However, the black-box nature of many such models allows only to quantify uncertainty through ad hoc superstructures, which require to develop and train a model in an uncertainty-aware fashion. However, for applications where previously trained models are already in operation, it would be interesting to develop uncertainty quantification approaches acting as lightweight "plug-ins" that can be applied on top of such models without modifying and re-training them. In this contribution we present a research activity of the Pattern Recognition and Applications Lab of the University of Cagliari related to a recently proposed *post hoc* uncertainty quantification method, we named *dropout injection*, which is a variant of the well-known Monte Carlo dropout, and does not require any re-training nor any further gradient descent-based optimization; this makes it a promising, lightweight solution for integrating uncertainty quantification on any *already-trained* neural network. We are investigating a theoretically grounded solution to make dropout injection as effective as Monte Carlo dropout through a suitable rescaling of its uncertainty measure; we are also evaluating its effectiveness in the computer vision tasks of crowd counting and density estimation for intelligent video surveillance, thanks to our participation in a project funded by the European Space Agency.

Keywords

Trustworthy AI, Uncertainty Quantification, Monte Carlo Dropout, Dropout Injection, Crowd Counting

1. Introduction

Today, Machine Learning (ML) and Deep Learning (DL) have acquired a prominent role in the Artificial Intelligence (AI) field. One of the main problems arising from their widespread adoption is the black-box nature of many ML and DL models. A related issue we are exploring in our ongoing work is the trustworthiness of their predictions. One possible approach to tackle this problem is the introduction of methods for quantifying the *uncertainty* of the model's predictions, which is fundamental in many critical applications (e.g., healthcare and public security) [1], to provide users with a well-rounded interpretation of the system's outputs and raise awareness of possible errors a DL-based system can make. Many methods based on principled Bayesian approaches have been proposed for this purpose, such as Monte-Carlo dropout [2] and ensemble [3] methods. A drawback of most such methods is that they require an ad hoc training process which also requires a higher processing cost during development; this can be impractical in application scenarios where previously trained models are already in operation. For this reason, at the Pattern Recognition and Applications Laboratory (PRALab) of the University of Cagliari[1] we are currently focusing on post hoc uncertainty quantification methods, which one can build upon already-trained neural networks. In particular, we are investigating a post hoc variant of Monte Carlo dropout that we named *dropout injection*, which has the additional advantage of not needing any further gradient descent-based optimization process, in contrast to other post hoc methods [4, 5]; this means it can potentially work as a zero-training cost plug-in for any already deployed DL-based system.

We are currently investigating practical applications of the post hoc injected dropout method to computer vision. In particular, we are focusing on the crowd counting and density estimation tasks related to intelligent video surveillance, which we addressed in our participation in the project IMMAGINA (IMaging MAnagement Guidelines and Informatics Network for law enforcement Agencies, 2020-2022), funded by the European Space Agency.

In this application context, uncertainty quantification can enable end users (e.g., law enforcement operators) to better interpret the outputs of DL-based tools, making them aware of unreliable estimates due to, e.g., dif-

Ital-IA 2023: 3rd National Conference on Artificial Intelligence, organized by CINI, May 29–31, 2023, Pisa, Italy
*Corresponding author.
emanuele.ledda@uniroma1.it (E. Ledda); lorenzo.putzu@unica.it (L. Putzu); rita.delussu@unica.it (R. Delussu); fumera@unica.it (G. Fumera); fabio.roli@unige.it (F. Roli)
0000-0003-2856-9381 (E. Ledda); 0000-0001-5361-8793 (L. Putzu); 0000-0002-6504-3779 (R. Delussu); 0000-0001-5300-226X (G. Fumera); 0000-0003-4103-9190 (F. Roli)

[1]https://pralab.diee.unica.it

ferent operating conditions than the ones represented in training data. We are also studying how to exploit uncertainty estimates provided by injected dropout to automatically improve the accuracy of an already trained crowd counting and density estimation model *online* (i.e., during operation).

2. Uncertainty in Neural Networks

Traditional neural networks are usually deterministic models that associate a point estimate with an input sample. However, such estimators are far from perfect, and their mistakes are often noteworthy: hence, developing tools devoted to capturing such uncertainty is essential in many applications [1].

Uncertainty sources, in neural networks, can be of two types [6]:

- **Epistemic Uncertainty**, caused by lack of knowledge about the correct model's parameters;
- **Aleatoric Uncertainty**, caused by the intrinsic randomness of the prediction.

Epistemic uncertainty can be, theoretically, reduced to zero by collecting enough data, while aleatoric uncertainty cannot [6]. For example, if one trains a neural network to predict the outcome of a dice roll no matter the number of training instances, the inherent randomness of the stochastic event makes correct outcomes with zero uncertainty impossible for *any* prediction model.

For aleatoric uncertainty quantification, a common choice is to define metrics upon the probability distribution $p(y|x,\theta)$ of the outcomes y given the input x and the model parameters θ, such as the entropy [7]. However, although obtaining such measures for classification problems is straightforward, further superstructures are necessary for regression [7].

For epistemic uncertainty quantification, on the other hand, the problem becomes more complicated: indeed, for obtaining such a measure in neural networks, a quantification of the model's parameters uncertainty is needed[2].

2.1. Bayesian Neural Networks

A well-known approach for dealing with prediction models' uncertainty is using Bayesian methods [8]; indeed, such models output a full probability distribution that is not only conditioned on the input but also dependent upon the model parameters distribution. Nevertheless, traditional neural networks fall into the non-Bayesian category, which means they cannot capture this dependence because of their deterministic set of parameters. One possible solution to make them able to estimate the uncertainty of their prediction is using their Bayesian extension, namely **Bayesian Neural Networks** [9, 10] (BNN). The output of a BNN can be considered as a predictive distribution conditioned on the input and the training data D:

$$p(y|x,D) = \int_\theta p(y|x,\theta) \cdot p(\theta,D)\mathrm{d}x, \tag{1}$$

where x is the input, y is the output, and θ are the parameters of the model. Although this formulation has interesting theoretical properties, obtaining the probability distribution of the model's parameters $p(\theta,D)$ is infeasible due to the many connections in modern neural networks. This problem is approached in the literature by substituting the distribution of the parameters $p(\theta,D)$ using an approximated distribution $q(\theta)$ [11].

Among the existing approximation strategies, **Monte Carlo Dropout** [2] is one of the most prominent in the context of BNNs. The *dropout* technique randomly deactivates the network neurons with a predefined probability φ. It was initially designed as a stochastic regularization technique [12], which means one should keep dropout active only at training time. However, when used as a Bayesian approximation, dropout is kept active also at inference time, combined with a set of Monte Carlo forward passes: this will lead to the construction of the desired approximated parameters distribution $q(\theta|\varphi)$. If one extracts a possible set of parameters θ_t from the distribution of the parameters $q(\theta|\varphi)$ at each step t of the T Monte Carlo forward passes, it is possible to approximate the predictive distribution as follows:

$$p(y|x,D) = \frac{1}{T}\sum_{t=1}^{T} p(y|x,\theta_t). \tag{2}$$

After computing the predictive distribution, one should choose a metric for modeling the predictive uncertainty. For classification problems, a categorical distribution is usually employed: for quantifying the model's uncertainty, typical choices are predictive entropy, predictive variance, or mutual information [13, 7]. For regression problems, instead, one possible option is to assume that the predictive distribution $p(y|x)$ follows a Normal distribution: in this context, the variance and standard deviations of the model output are the typical choices for modeling uncertainty [7].

2.2. Dropout Injection

It is essential to note that in the original formulation of dropout as a Bayesian approximation [2], dropout is intended to be activated both for the training and the

[2]Neural networks are usually assumed to be capable of learning any function, so the lack of knowledge of the correct model parameters is usually considered a proxy for epistemic uncertainty.

testing phase. However, some recent work proposed using Monte Carlo dropout on arbitrary, *already-trained* neural networks, regardless of their use of dropout at training time [14].

Our ongoing work started from the observation that such modification can be advantageous compared to the original version of Monte Carlo dropout because of its flexibility (searching from a suitable dropout rate does not require multiple training), its non-invasivity (acting as a plug-in), and its lightness (no ad hoc training is needed). However, we noticed that this variant of Monte Carlo dropout has never been comprehensively analyzed or compared with its original version, neither by the author proposing this modification [14] nor by any following work using this technique. Taking these considerations as a starting point, we began our research by analyzing this technique, which we call **dropout injection**, and its main difference from the original version, which we call **embedded dropout**.

The first characteristic we observed was the necessity of suitably rescaling the uncertainty measure when using dropout injection [15]. Indeed, it turned out that such a problem, which is also present to a less extent in embedded dropout, is amplified by injecting dropout because a network trained without dropout is less robust to random neural deactivation (since it has not been optimized for handling such a circumstance). Without a suitable rescaling, if the injected dropout rate is too high, it may result in low-quality prediction; on the other hand, if it is too low, it will result in almost no variations in the network output, which would be ineffective for uncertainty estimation. We found that a possible solution consists of suitably scaling the uncertainty measure. To this aim, we exploited a technique originally proposed as a post hoc calibration method for embedded dropout, named "σ-scaling" [16]. Such a scaling technique can be applied to jointly seek a suitable scaling factor and dropout rate in a unified optimization problem with no additional processing cost with respect to the original one. To this aim, instead of simply minimizing the negative log-likelihood (NLL) – that, in variational inference, is used as a proxy for the divergence between the actual and the approximated distribution [11] – we consider a proper scaling factor, which can be computed analytically [16] by using a validation set. This results in finding the desired dropout rate Φ among all possible values φ, which minimizes:

$$\Phi = \arg\min_{\varphi} \frac{1}{N} \sum_{n=1}^{N} \frac{1}{2} \frac{(y_n - \hat{y}_n(\varphi))^2}{C(\varphi) \cdot \hat{\sigma}_n^2(\varphi)} + \frac{1}{2} \log(C(\varphi) \cdot \hat{\sigma}_n^2(\varphi)) \,, \tag{3}$$

where N is the size of the validation set, $C(\varphi)$ is the optimal scale factor, and, finally, y_n, $\hat{y}_n(\varphi)$ and $\hat{\sigma}_n^2(\varphi)$ are respectively the ground truth, the model's prediction and the uncertainty of the n^{th} sample of the validation set.

With this adaptation, we have obtained well-calibrated uncertainty measures while keeping relatively low dropout rates (which, when using injected dropout, is crucial to avoid damaging the prediction quality).

3. Trustworthy Intelligent Video Surveillance

In this section we show how our implementation of dropout injection can be practically applied to quantify the uncertainty of state-of-the-art DL architectures for crowd counting and density estimation, and how uncertainty measures can also be exploited to improve their accuracy.

3.1. Crowd Counting and Density Estimation

Crowd counting and density estimation are computer vision (CV) tasks aimed at estimating the number of people present in images or video frames and their density map. By their nature, the technologies employed for solving these tasks evolved together with image analysis and processing, and nowadays, DL models represent the predominant choice. DL-based crowd counting models can be categorized into two main approaches [17]:

- **detection-based** models, which rely on object detectors for counting the number of people;
- **regression-based** models, which first perform a multi-variate regression to estimate the crowd density map, whose sum of pixel values corresponds to the people count.

In our research, we focus on regression-based approaches. In this context, the problem can be viewed as an image-to-image translation problem [20], where input and output consist of a crowd image and a crowd density map. The ground truth density map of a training image is typically obtained by first manually annotating the head position of each pedestrian. A binary map is then constructed, containing zero values for each pixel except for the ones corresponding to the head's locations, which are set to one. The density map is finally obtained by applying to the binary map a convolution with a kernel with unit area, e.g., a Gaussian (see the example in Fig. 1). By construction, the ground truth crowd count equals the sum of all the pixels of the resulting density map. During inference, a trained model estimates the density map of input images, and the corresponding estimate of the crowd count is obtained by summing up its pixel values.

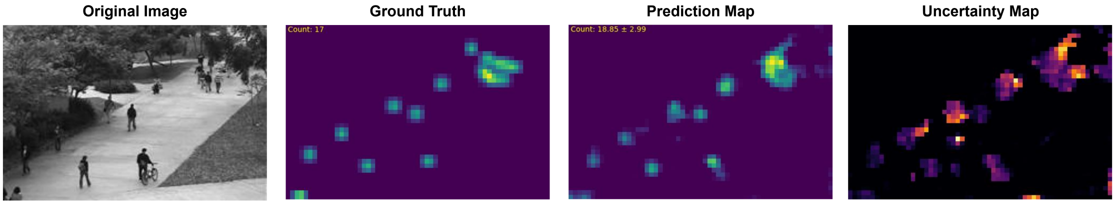

Figure 1: From left to right: an example of crowd image from the UCSD [18] benchmark data set, the corresponding ground truth density map computed using head location annotations, the density map predicted by the state-of-the-art, pre-trained Multi-Column Neural Network (MCNN [19]) model, and the uncertainty map obtained by injecting dropout on MCNN.

3.2. Uncertainty Quantification for Crowd Counting

State-of-the-art DL architectures for regression-based crowd counting and density estimation do not incorporate any measure of uncertainty. To this aim, dropout injection can be implemented during inference to obtain a full probability distribution for each pixel of the estimated density map, which is modeled with a Normal distribution: its mean represents the point prediction, and we employ its variance as a measure of uncertainty. As shown in the example of Fig. 1, this approach provides an estimated crowd density map and a corresponding uncertainty map. We obtain the total variance on the predicted crowd count by summing up all the pixel values of the uncertainty map. This allows us to compute a confidence interval on the predicted crowd count. Our first experiments provided evidence that the accuracy of the confidence intervals obtained through our implementation of dropout injection relies on a suitable scaling of the underlying uncertainty measure. It is worth pointing out that a pixel-wise uncertainty measure makes it possible to compute confidence intervals not only for the whole image but also for sub-regions of it. This feature is helpful when a video surveillance operator is interested in monitoring a crowd in a specific region of a video.

3.3. Improving Crowd Counting Accuracy through Uncertainty Quantification

In our ongoing work, we found that dropout injection could also be helpful for automatically detecting image regions with no pedestrians but with non-zero estimated density, which we call “false-positives;” in particular, from our experiments, it seems that such regions are characterized by relatively low density values and relatively high uncertainty. An example is shown in Fig. 2. Detecting false positive regions would allow us to automatically reject their contribution to the predicted crowd count and to the corresponding confidence interval, thus improving the accuracy of the underlying model. We are currently investigating a suitable policy for false positive detection and correction.

3.4. The IMMAGINA Project

The crowd counting and density estimation tasks, that we used to investigate the injected dropout technique, were chosen as they were the subjects of previous research activities by the PRA Lab, involving also the participation in research projects. The most recent project is IMMAGINA[3] (IMaging MAnagement Guidelines and Informatics Network for law enforcement Agencies), funded by the European Space Agency under the ARTES Integrated Applications Promotion Programme (Nov. 2020 – Oct. 2022), aimed at exploring applications integrating space assets and 5G networks, including law enforcement. Our task was to develop a high-TRL prototype of a real-time crowd counting and density estimation system, in the form of a Web application service accessible through a Web browser both from the control room of a law enforcement agency (LEA) and from mobile devices (tablets) of officers in the field. Our participation in IMMAGINA helped us to better understand the needs of a relevant category of potential end users (LEA operators) of AI- and machine learning-enabled computer vision tools, whose trustworthiness is crucial for their acceptance in critical, security-related operational scenarios.

4. Conclusions and Future Work

In this contribution, we summarized our ongoing research on uncertainty quantification related to injected dropout, a post hoc version of the well-known Monte Carlo dropout technique. Injected dropout allows incorporating uncertainty quantification on already-trained neural networks. We are currently investigating its application to security-related computer vision tasks, focusing on regression-based crowd counting and density estimation methods using deep learning models, thanks to our participation in the recent IMMAGINA project. In this

[3] https://business.esa.int/projects/immagina

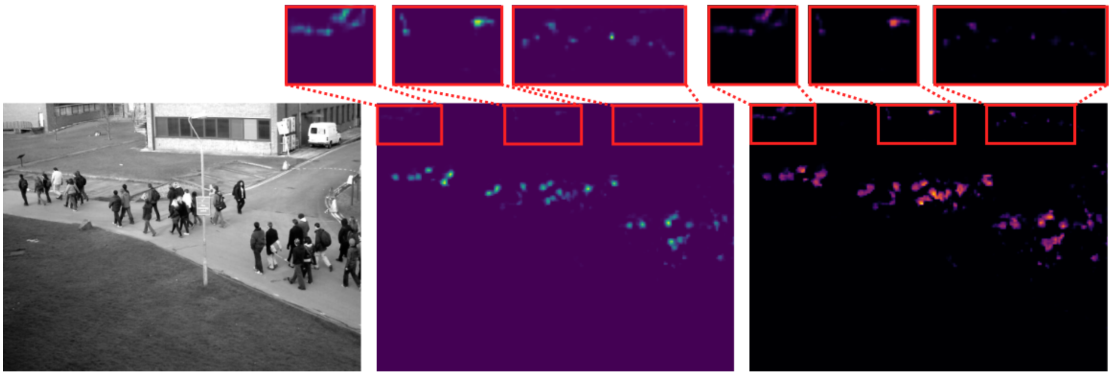

Figure 2: An example of "false positive" pedestrian regions from the PETS2009 [21] benchmark data set, that could be detected through the uncertainty map. From left to right: the original frame, the predicted density map, and the uncertainty map. Three different false positive regions are shown (magnified for better visualization), corresponding to relatively low density values and relatively high uncertainty.

kind of task, an easy-to-interpret representation of the uncertainty on the outputs of a machine learning model, besides its bare point prediction, could be beneficial for end users (e.g., LEA operators monitoring crowds in public spaces), both to better support their decisions and to improve their trust in machine learning-based systems. In particular, we showed how injected dropout allows to compute a pixel-level uncertainty map associated to the predicted density map, and a confidence interval on the corresponding predicted crowd count.

Looking ahead, based on preliminary empirical evidence, we are investigating the possibility of automatically improving the accuracy of the predicted density map and crowd count by exploiting pixel-based uncertainty quantification, which seems capable of highlighting "false positive" pedestrian detections in the density map. Other interesting directions for future research include the extension of the scope of our investigation to further post hoc uncertainty quantification methods, their application to classification with a reject option, and the detection of out-of-distribution as well as adversarial examples.

Acknowledgments

This work was partially supported by the projects: "Law Enforcement agencies human factor methods and Toolkit for the Security and protection of CROWDs in mass gatherings" (LETSCROWD), Horizon 2020 programme, grant agreement No. 740466, funded by the EU; "IMaging MAnagement Guidelines and Informatics Network for law enforcement Agencies" (IMMAGINA), ARTES Integrated Applications Promotion Programme, contract No. 4000133110/20/NL/AF, funded by the ESA; "Science and engineering Of Security of Artificial Intelligence" (S.O.S. AI) included in the Spoke 3 – Attacks and Defences of the Research and Innovation Program PE00000014, "SEcurity and RIghts in the CyberSpace (SERICS)", National Recovery and Resilience Plan, Mission 4 "Education and Research" - Component 2 "From Research to Enterprise" - Investment 1.3, funded by the EU – NextGenerationEU; ELSA – European Lighthouse on Secure and Safe AI, grant agreement No. 101070617 funded by the EU. E. Ledda is affiliated with the Italian National PhD in Artificial Intelligence, Sapienza University of Rome. He acknowledges the cooperation with and the support from the PRA Lab of the University of Cagliari.

References

[1] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. W. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, V. Makarenkov, S. Nahavandi, A review of uncertainty quantification in deep learning: Techniques, applications and challenges, Inf. Fusion 76 (2021) 243–297. doi:10.1016/j.inffus.2021.05.008.

[2] Y. Gal, Z. Ghahramani, Dropout as a bayesian approximation: Representing model uncertainty in deep learning, in: M. Balcan, K. Q. Weinberger (Eds.), Proc. of the 33nd Int. Conf. on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016, volume 48 of *JMLR Workshop and Conf. Proc.*, JMLR.org, 2016, pp. 1050–1059. URL: http://proceedings.mlr.press/v48/gal16.html.

[3] B. Lakshminarayanan, A. Pritzel, C. Blundell, Simple and scalable predictive uncertainty estimation using deep ensembles, in: I. Guyon, U. von Luxburg,

S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, R. Garnett (Eds.), Advances in Neural Inf. Proc. Systems 30, 2017, December 4-9, 2017, Long Beach, CA, USA, 2017, pp. 6402–6413. URL: https://proceedings.neurips.cc/paper/2017/hash/9ef2ed4b7fd2c810847ffa5fa85bce38-Abstract.html.

[4] A. Kristiadi, M. Hein, P. Hennig, Learnable uncertainty under laplace approximations, in: C. P. de Campos, M. H. Maathuis, E. Quaeghebeur (Eds.), Proc. of the 37th Conference on Uncertainty in Artificial Intelligence, UAI 2021, Virtual Event, 27-30 July 2021, volume 161 of *Proc. of Machine Learning Research*, AUAI Press, 2021, pp. 344–353. URL: https://proceedings.mlr.press/v161/kristiadi21a.html.

[5] U. Upadhyay, S. Karthik, Y. Chen, M. Mancini, Z. Akata, Bayescap: Bayesian identity cap for calibrated uncertainty in frozen neural networks, in: S. Avidan, G. J. Brostow, M. Cissé, G. M. Farinella, T. Hassner (Eds.), Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proc., Part XII, volume 13672 of *Lecture Notes in Computer Science*, Springer, 2022, pp. 299–317. doi:10.1007/978-3-031-19775-8_18.

[6] Y. Gal, Uncertainty in Deep Learning, Ph.D. thesis, University of Cambridge, 2016.

[7] A. Kendall, Y. Gal, What uncertainties do we need in bayesian deep learning for computer vision?, in: I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, R. Garnett (Eds.), Advances in Neural Inf. Proc. Systems 30, 2017, December 4-9, 2017, Long Beach, CA, USA, 2017, pp. 5574–5584. URL: https://proceedings.neurips.cc/paper/2017/hash/2650d6089a6d640c5e85b2b88265dc2b-Abstract.html.

[8] E. Hüllermeier, W. Waegeman, Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods, Mach. Learn. 110 (2021) 457–506. doi:10.1007/s10994-021-05946-3.

[9] C. Blundell, J. Cornebise, K. Kavukcuoglu, D. Wierstra, Weight uncertainty in neural network, in: F. R. Bach, D. M. Blei (Eds.), Proc. of the 32nd Int. Conf. on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015, volume 37 of *JMLR Workshop and Conf. Proc.*, JMLR.org, 2015, pp. 1613–1622. URL: http://proceedings.mlr.press/v37/blundell15.html.

[10] D. J. C. MacKay, A practical bayesian framework for backpropagation networks, Neural Comput. 4 (1992) 448–472. doi:10.1162/neco.1992.4.3.448.

[11] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, L. K. Saul, An introduction to variational methods for graphical models, Mach. Learn. 37 (1999) 183–233. doi:10.1023/A:1007665907178.

[12] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (2014) 1929–1958. doi:10.5555/2627435.2670313.

[13] L. Smith, Y. Gal, Understanding measures of uncertainty for adversarial example detection, in: A. Globerson, R. Silva (Eds.), Proc. of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI 2018, Monterey, California, USA, August 6-10, 2018, AUAI Press, 2018, pp. 560–569. URL: http://auai.org/uai2018/proceedings/papers/207.pdf.

[14] A. Loquercio, M. Segù, D. Scaramuzza, A general framework for uncertainty estimation in deep learning, IEEE Robotics Autom. Lett. 5 (2020) 3153–3160. doi:10.1109/LRA.2020.2974682.

[15] E. Ledda, G. Fumera, F. Roli, Dropout injection at test time for post hoc uncertainty quantification in neural networks, CoRR abs/2302.02924 (2023). doi:10.48550/arXiv.2302.02924. arXiv:2302.02924.

[16] M. Laves, S. Ihler, J. F. Fast, L. A. Kahrs, T. Ortmaier, Well-calibrated regression uncertainty in medical imaging with deep learning, in: T. Arbel, I. B. Ayed, M. de Bruijne, M. Descoteaux, H. Lombaert, C. Pal (Eds.), Int. Conf. on Medical Imaging with Deep Learning, MIDL 2020, 6-8 July 2020, Montréal, QC, Canada, volume 121 of *Proc. of Machine Learning Research*, PMLR, 2020, pp. 393–412. URL: http://proceedings.mlr.press/v121/laves20a.html.

[17] G. Gao, J. Gao, Q. Liu, Q. Wang, Y. Wang, Cnn-based density estimation and crowd counting: A survey, CoRR abs/2003.12783 (2020). URL: https://arxiv.org/abs/2003.12783. arXiv:2003.12783.

[18] A. B. Chan, Z. J. Liang, N. Vasconcelos, Privacy preserving crowd monitoring: Counting people without people models or tracking, in: 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), 24-26 June 2008, Anchorage, Alaska, USA, IEEE Computer Society, 2008. doi:10.1109/CVPR.2008.4587569.

[19] Y. Zhang, D. Zhou, S. Chen, S. Gao, Y. Ma, Single-image crowd counting via multi-column convolutional neural network, in: 2016 IEEE Conference on Comp. Vis. and Patt. Rec., CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, IEEE Computer Society, 2016, pp. 589–597. doi:10.1109/CVPR.2016.70.

[20] P. Isola, J. Zhu, T. Zhou, A. A. Efros, Image-to-image translation with conditional adversarial networks, in: 2017 IEEE Conf. on Comp. Vis. and Patt. Rec., CVPR 2017, Honolulu, HI, USA, July 21-26, 2017, IEEE Computer Society, 2017, pp. 5967–5976. doi:10.1109/CVPR.2017.632.

[21] J. M. Ferryman, A. Shahrokni, Pets2009: Dataset and challenge, 2009 Twelfth IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance (2009) 1–6.