Risk-aware Real-time Object Detection

Misael Alpizar Santana
Department of Computer Science
University of York, York, UK
misael.alpizarsantana@york.ac.uk

Radu Calinescu
Department of Computer Science
University of York, York, UK
radu.calinescu@york.ac.uk

Colin Paterson
Department of Computer Science
University of York, York, UK
colin.paterson@york.ac.uk

***Abstract*—Autonomous systems such as self-driving cars and infrastructure inspection robots must be able to mitigate risk by dependably detecting entities that represent factors of risk in their environment (e.g., humans and obstacles). Nevertheless, current machine learning (ML) techniques for real-time object detection disregard risk factors in their training and verification. As such, they produce ML models that place equal emphasis on the correct detection of *all* classes of objects of interest—including, for instance, buses and cats in a self-driving scenario. To address this limitation of existing solutions, this short paper introduces a work-in-progress method for the development of *risk-aware ML ensembles* for real-time object detection. Our new method supports the dependable use of real-time object detection in autonomous systems by (i) identifying the risks that require treatment, (ii) training a set of ML models that mitigate these risks, and (iii) using multi-objective genetic algorithms to combine the ML models into risk-aware ML ensembles. We present preliminary experiments that show the effectiveness of our method at constructing a dependable ML ensemble for real-time object detection in a simulated self-driving case study.**

***Index Terms*—object detection, risk, risk mitigation, ensembles**

I. Introduction

Object detection [18], [19] is a computer vision task with applications ranging from pedestrian detection in autonomous driving to face recognition in smart photography. This complex task involves detecting objects of interest and their position in an image, and is typically performed using machine learning (ML), computer vision techniques, or a combination thereof. When applied online, e.g., to the successive frames of a video stream, the task is termed *real-time object detection* [11], [14].

Our paper focuses on the dependable use of ML-based real-time object detection (RTOD) in safety-critical applications such as autonomous driving. Despite significant advances, in particular in the area of deep-learning, RTOD cannot be 100% accurate due to challenges ranging from insufficient training data [7] and class imbalance [8] to inherent localisation and identification errors [11], [17]. As such, the use of RTOD in safety-critical applications introduces risks that need to be systematically assessed and mitigated. To the best of our knowledge, existing RTOD solutions disregard this need. In particular, they place equal emphasis on the detection accuracy of all classes of objects of interest—including, for example, buses, cats, bikes and birds in an autonomous driving scenario.

In this paper, we introduce a work-in-progress method that addresses this major limitation of current RTOD solutions. To that end, we use (i) a risk management process recommended by the ISO 31010 standard [6] to identify high-risk RTOD misclassifications, (ii) a risk-aware deep learning technique to produce ML models that mitigate each of these misclassifications, and (iii) a multi-objective genetic algorithm to combine the resulting ML models into a *risk-aware RTOD ensemble*. Techniques (i)–(iii) and the evaluation of our method for the widely used VOC2012 object detection data set [3] form the main contributions of the paper.

II. Background

Object detection [2], [10], [18] is defined as a function that maps an image to a list of objects $O = (o_1, o_2, \ldots, o_n)$, where the i-th detected object $o_i = (c_i, box_i)$ specifies:

- (estimate) probabilities $c_i = (c_{i1}, c_{i2}, \ldots, c_{iN})$ that object i belongs to each of N classes of interest, $\sum_{j=1}^{N} c_{ij} = 1$;
- a "bounding box" box_i comprising the coordinates of the top-left and bottom-right corners of the image region where object i is located.

ML-based RTOD solutions such as YOLO [11], [13], SSD [9] and RetinaNet [8] realise this function by means of a complex multi-stage process:

1) In a first stage, a feature-extraction neural network (NN) is used to detect potential objects in the input image.
2) In a second, *detection* stage, *anchor bounding boxes* (i.e., approximate bounding boxes drawn from a set of pre-defined box sizes) are fitted around these objects, and then adjusted appropriately. Next, an NN classifier is used to estimate the probabilities that the object within each bounding box belongs to the N classes of interest, as well as a confidence measure that an object is actually present in each of these boxes.
3) In a final stage, *bounding box matching and labelling* is performed. To start with, irrelevant and duplicate boxes are eliminated. A box is deemed irrelevant if the product between its confidence measure and maximum class probability is below a predefined "relevance" threshold. To identify duplicates, a *non-maximum suppression* algorithm processes the remaining boxes in decreasing probability order, i.e., starting with the box i with the highest class probability $c_{i,j}$, and eliminating all unprocessed boxes that overlap significantly with box i. The overlap between two boxes is deemed significant if their *intersection over union* (IoU) measure (i.e., the ratio between their intersection and

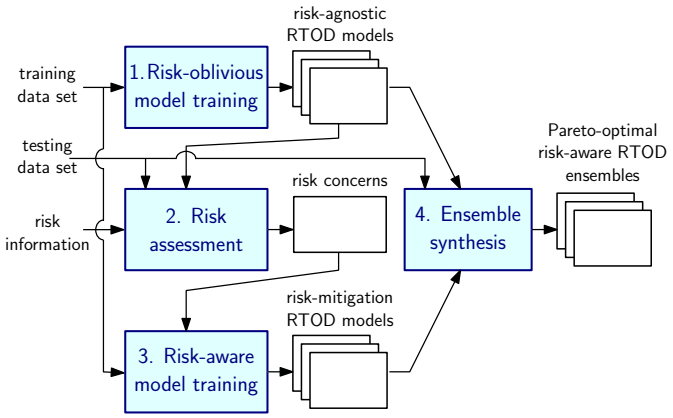

Fig. 1: Risk-aware RTOD ensemble synthesis method

their union) exceeds a predefined threshold. Each remaining box is then labelled with the name corresponding to its maximum class probability.

The NNs employed by ML-based RTOD solutions are trained and tested using large sets of image samples annotated with both the bounding boxes and the class labels for all relevant objects in each image. For detailed descriptions of ML-based RTOD, we refer the reader to [8], [9], [11], [13].

III. Approach

Our method for the synthesis of ML ensembles for risk-aware RTOD (Fig. 1) comprises the four steps detailed below.

Step 1: Risk-oblivious model training. In this step, we use standard ML training to generate a set of RTOD models. These models are obtained from the same *training data set*, and different random initial weights for the RTOD NNs. In this way, we avoid ending up with a single, low-accuracy RTOD model due to an unfavourable selection of random initial weight values. The models obtained in this step are *risk-oblivious* because the *loss function* used in standard NN training weighs the misclassification of class $i1$ as class $i2 \neq i1$ as equally undesirable irrespective of what the two classes represent in the real world.

Step 2: Risk assessment. This step uses a five-point semi-quantitative risk assessment technique from the ISO 31010 standard [6] to identify *risk concerns*, i.e., RTOD object misclassifications that induce unacceptably high risks. This assessment is underpinned by the following *risk information*, which needs to be obtained from domain experts:

1) The impact level $imp(i,j) \in \{\mathsf{VL}, \mathsf{L}, \mathsf{M}, \mathsf{H}, \mathsf{VH}\}$ of misclassifying an object belonging to class i as an object from class $j \neq i$, for all $1 \leq i, j \leq N$.[1]
2) The likelihood $l_e(i) \in \{\mathsf{VL}, \mathsf{L}, \mathsf{M}, \mathsf{H}, \mathsf{VH}\}$ of encountering an object of class i in the environment where the RTOD model will be used, for all $1 \leq i \leq N$.
3) The likelihood of misclassification thresholds $0 = l_m^0 < l_m^{\mathsf{VL}} < l_m^{\mathsf{L}} < l_m^{\mathsf{M}} < l_m^{\mathsf{H}} < l_m^{\mathsf{VH}} = 1$, with the likelihood of misclassifying class i objects as class $j \neq i$ objects deemed to be very low, low, medium, high, and very high when the fraction $mis(i,j)$ of *test data set* objects of class i misclassified as objects of type j satisfies $mis(i,j) \in [0, l_m^{\mathsf{VL}}]$, $mis(i,j) \in (l_m^{\mathsf{VL}}, l_m^{\mathsf{L}}]$, $mis(i,j) \in (l_m^{\mathsf{L}}, l_m^{\mathsf{M}}]$, $mis(i,j) \in (l_m^{\mathsf{M}}, l_m^{\mathsf{H}}]$ and $mis(i,j) \in (l_m^{\mathsf{H}}, 1]$, respectively.
4) The maximum acceptable risk level $arl \in \{\mathsf{VL}, \mathsf{L}, \mathsf{M}, \mathsf{H}\}$ for the application using the RTOD model ($arl = \mathsf{VH}$ is not used, as it would make all risks being acceptable).

[1] VL=very low, L=low, M=medium, H=high, VH=very high (cf. [6])

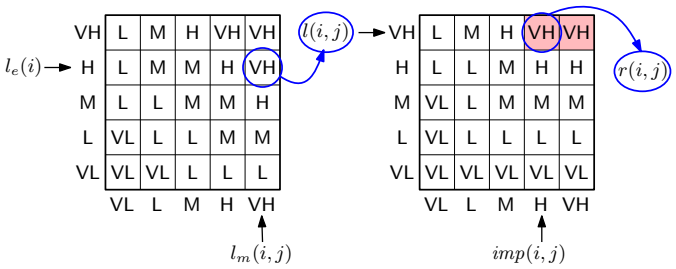

Fig. 2: Calculation of the risk level $r(i,j)$ for misclassifying class i as class j given the likelihood of encounter $l_e(i)$ for class i and the impact $imp(i,j)$ of such a misclassification, where $l_m(i,j)$ is the calculated likelihood of misclassification.

Given this risk information, we consider each pair of classes $i \neq j$, and we calculate $mis(i,j)$ as the mean fraction of misclassifications of class i objects as class j objects over the risk-oblivious RTOD models from Step 1. Next, we use the thresholds $l_m^{\mathsf{VL}}, l_m^{\mathsf{L}}, l_m^{\mathsf{M}}$ and l_m^{H} to establish the likelihood $l_m(i,j) \in \{\mathsf{VL}, \mathsf{L}, \mathsf{M}, \mathsf{H}, \mathsf{VH}\}$ of class i being misclassified as class j by a risk-oblivious RTOD model. Finally, we use likelihood and risk matrices [6] to establish an overall likelihood $l(i,j)$ and a risk level $r(i,j)$ for the misclassification of class i as class j as shown in Fig. 2. The shading of the 'VH' elements from the risk matrix indicates that the diagram assumes a maximum acceptable risk level $arl = \mathsf{H}$, and that any pair of classes (i,j) whose misclassification risk level resides in the shaded area represents a *risk concern*, i.e., a risk that needs to be mitigated. The outcome of Step 2 of our method is a set of all such risk concerns for the RTOD models.

Step 3: Risk-aware model training. If risk concerns were identified in Step 2, this step produces a configurable number of *risk-mitigation RTOD models* for each risk concern (i,j). These are models whose NNs are trained using a loss function $\mathcal{L}(\theta)$ that prioritises the minimisation of misclassifying class i as class j over that of all other misclassifications:

$$\mathcal{L}(\theta) = -\frac{1}{M} \sum_{k=1}^{M} \sum_{l=l}^{N} w(y_k, \hat{y}_k) y_{kl} \log p_{kl}, \qquad (1)$$

where θ represents the NN weights to be learnt, M is the number of samples in the training data set, $y_{kl} = 1$ if the true class y_k for the k-th sample is $y_k = l$ and $y_{kl} = 0$ otherwise, p_{kl} is the value of the l-th NN output neuron for the k-th sample, and (unique to our method) $w(y_k, \hat{y}_k) > 0$ is a weight associated with classifying sample k as class $\text{argmax}_{l=1}^{N} p_{kl} =$

$\hat{y}_k$; to obtain RTOD models that mitigate the risk concern (i, j), we use:

$$w(y_k, \hat{y}_k) = \begin{cases} \omega N^2/(N^2+\omega-1), & \text{if } y_k = i \wedge \hat{y}_k = j \\ N^2/(N^2+\omega-1), & \text{otherwise} \end{cases} \quad (2)$$

where $\omega > 1$ is a parameter of our risk-aware model training. Note that (i) setting $\omega = 1$ reduces (1) to the standard loss function used in NN training, and (ii) the use of $\omega > 1$ increases the contribution of the loss term $y_{kl} \log p_{kl}$ from (1) for samples k of class i misclassified as class j.

We show in Section IV that using the loss function (1) can yield RTOD models with significantly reduced misclassification rates $mis(i, j)$. Of course, this technique cannot guarantee that such models will be obtained under all circumstances. For instance, the technique may be unable to mitigate risks due to training data sets that are unbalanced or too small, or poorly chosen NN architectures. In such cases, the issue that led to the risk concern needs to be fixed, or it may be possible to mitigate the risk by reducing its impact (e.g., a self-driving car can drive slower) or likelihood of encounter (e.g., by banning bicycles on certain roads used by self-driving cars).

Step 4: Ensemble synthesis. In this step, we select a subset of risk-oblivious models from Step 1 and risk-mitigating models from Step 3, and use a multi-objective genetic algorithm (GA) to optimise a set of weights for combining these models into ensembles that achieve Pareto-optimal trade-offs between:

1) maximising the *F1 score*, an established performance measure of ML models [10];
2) minimising the residual risk

$$residualRisk = \sum_{\substack{1 \le i,j \le N \\ qr(i,j) > qarl}} [qr(i, j) - q(arl)], \quad (3)$$

where $q(arl)$ and $qr(i, j)$ are quantitative variants of arl and $r(i, j)$, respectively. The former is defined by $q(\mathsf{VL}) = 1$, $q(\mathsf{L}) = 2$, etc. The latter is given by $qr(i, j) = q(r(i, j)) - 1 + \frac{miss(i,j) - l_m^{pred(r(i,j))}}{l_m^{r(i,j)} - l_m^{pred(r(i,j))}}$, where $pred(r(i, j))$ is the predecessor of $r(i, j)$: $pred(\mathsf{VL}) = 0$, $pred(\mathsf{L}) = \mathsf{VL}$, etc.

The weights $(W_{ji})_{1 \le j \le m, 1 \le i \le N}$, optimised by the GA are those used to combine the lists of objects detected by $m > 1$ RTOD models into a single list. This combination is carried out by first identifying sets of objects with significantly overlapping bounding boxes (according to the IoU measure, cf. Section II). For each such set S, which comprises at most one object per model (remember that significantly overlapping objects from the same model are eliminated cf. Section II), a single object is included in the ensemble. This object has:

- bounding box coordinates computed as the mean coordinates of the bounding boxes for the objects in S;
- class given by $\text{argmax}_{i=1}^{N} \sum_{j \in J} W_{ji} c_i^j$, where $J \subseteq \{1, 2, \ldots, m\}$ is the subset of models that contribute objects to S, $W_{ji} > 0$ is a weight that reflects the ability of j-th model to detect objects of class i, and c_j^j is model j's estimate probability that the object detected is of class i.

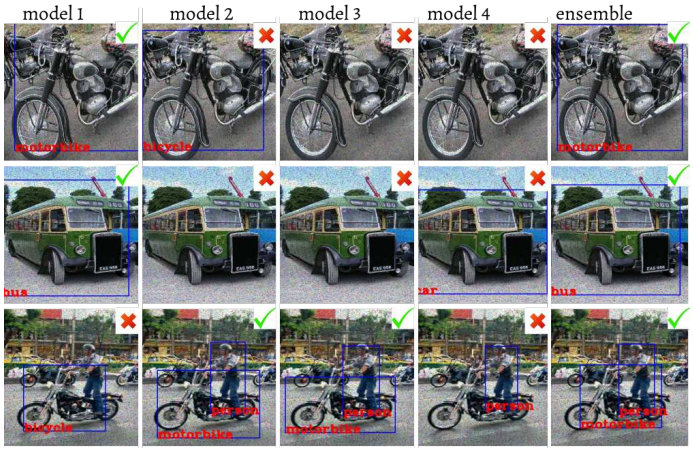

Fig. 3: Object detection comparing four models to the risk-aware RTOD ensemble synthesised from them.

IV. Preliminary Evaluation

To evaluate our method, we performed preliminary experiments using the VOC2012 object detection data set [3], which comprises 11,549 annotated images with objects from $N = 20$ classes ('person' and several types of vehicles, animals, etc.). We augmented this data set to 23,292 images using Gaussian and Pepper noise, and used 75% of these images for training and the rest for testing. We selected the RTOD risk information for the 20 VOC2012 classes to correspond roughly to a self-driving car operating in an urban setting, with $arl = \mathsf{H}$. The experimental results are summarised below and detailed in our supplementary material at https://rb.gy/lfk41w.

In Step 1, we built 30 risk-oblivious models using a version of the YOLOv3 RTOD which was implemented using TensorFlow 2.0 [10]. The risk assessment of these models in Step 2 produced three risk concerns: a) motorbike predicted as car; b) motorbike predicted as bicycle; and c) bus predicted as car.[2] Each risk-oblivious model was affected by at least one of these concerns, and the F1 score for the risk-oblivious models was in the range 0.56 to 0.59. In Step 3, we trained 21 risk-mitigation models for each risk concern, with seven such models built for each $\omega \in \{2, 5, 10\}$ in (2).

In Step 4, we synthesised Pareto-optimal RTOD ensembles using the DEAP evolutionary framework [4]. Eight models in total were passed to the ensemble synthesis: two risk-oblivious models and two risk-mitigation models for each concern, giving four types of models. These models were chosen randomly from the F1/residual-risk Pareto front for each type of models. From this model set, the GA was allowed to choose four models to use. We stopped the synthesis after 50 iterations, with each generation taking approximately 35 minutes on a desktop workstation. Fig. 3 illustrates the performance of a synthesised ensemble and its constituent models when applied to three images from the data set. A tick in the top left corner of the image indicates the model correctly identified all objects, whilst a cross indicates that at

[2]Due to space restrictions the values used to parameterise the problem are provided in our supplementary material at https://rb.gy/lfk41w.

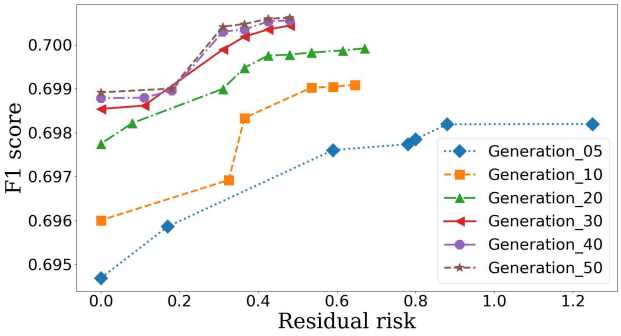

Fig. 4: Pareto fronts of RTOD ensembles generated by the GA

TABLE I: The mean RTOD time to process a single image in a batch of 100 images grows linearly with the ensemble size.

Ensemble size	1	2	4	6	8	16
RTOD time (s)	0.137	0.267	0.527	0.776	1.036	2.073

least one object was not detected or was misclassified. We can see that the ensemble is able to successfully identify objects even when the majority of models are unable to do so.

Fig. 4 shows the Pareto-optimal ensembles found by the GA. As the number of generations increases, the residual risk decreases and the F1 score improves, with diminishing returns as more GA generations are produced. All ensembles achieved much better F1 scores than the risk-oblivious models (0.6946 or higher compared to under 0.59 for the best risk-oblivious model). As expected, additional F1 performance can be traded against risk by selecting an ensemble from the Pareto front with a different set of trained weights.

To evaluate the scalability of our method, we examined the times taken to process an image for ensembles with increasing numbers of models. These times (Table I), obtained on a desktop workstation, indicate that our method is of practical use, and can be reduced further using the optimisations from [12].

V. Related work

Several existing RTOD solutions use ensembles of ML models [1], [2], [5], [15], [16]. Unlike our method, most of these solutions [2], [5] combine the output of their component ML models with equal weights, disregarding the fact that each model may be better at predicting certain classes. While the RTOD ensembles proposed by [1] and [15] use weighting in combining the outputs of their ML models, this weights are determined using a basic heuristic that is focused on improving accuracy by constructing averaged boxes, without explicitly reducing the number of misclassifications. In contrast, our method uses a multi-objective genetic algorithm that yields RTOD ensembles with Pareto-optimal trade-offs between object detection accuracy and risk. Finally, the RTOD ensemble generation solution devised by [16] uses a genetic algorithm to optimally combine the ML models from the ensemble. However, this solution focuses exclusively on optimising the ensemble accuracy, and therefore does not consider the risks associated with different misclassifications like our method. To the best of our knowledge, no existing RTOD approach considers the risks corresponding to different misclassifications.

VI. Conclusion

We introduced a new method for the synthesis of risk-aware ML ensembles for real-time object detection. The experimental results presented in the paper suggest that our method can effectively mitigate risk, supporting the development of dependable RTOD-based systems for safety-critical applications. Further evaluation for additional data sets and scenarios is required to confirm these preliminary findings, and to help refine the steps of the method.

Acknowledgements This project has received funding from the UKRI project EP/V026747/1 'Trustworthy Autonomous Systems Node in Resilience', and the Assuring Autonomy International Programme.

References

[1] Berat Mert Albaba and Sedat Ozer. SyNet: An ensemble network for object detection in UAV images. In *25th Intl. Conf. on Pattern Recognition*, pages 10227–10234, 2021.

[2] Ángela Casado-García and Jónathan Heras. Ensemble methods for object detection. In *ECAI 2020*, pages 2688–2695. 2020.

[3] Mark Everingham and John Winn. The PASCAL visual object classes challenge 2012 (VOC2012) development kit. *Pattern Analysis, Statistical Modelling and Computational Learning, Tech. Rep.*, 8:5, 2011.

[4] Félix-Antoine Fortin, François-Michel De Rainville, Marc-André Gardner, Marc Parizeau, and Christian Gagné. DEAP: Evolutionary algorithms made easy. *Journal of Machine Learning Research*, 13:2171–2175, jul 2012.

[5] Jing Gao et al. A general framework for mining concept-drifting data streams with skewed distributions. In *Proceedings of the 2007 SIAM Intl. Conf. Data Mining*, pages 3–14, 2007.

[6] ISO/IEC Standard 31010:2019. Risk management—risk assessment techniques, 2019.

[7] Tsung-Yi Lin et al. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*, pages 740–755, 2014.

[8] Tsung-Yi Lin et al. Focal loss for dense object detection. In *IEEE International Conference on Computer Vision*, pages 2980–2988, 2017.

[9] Wei Liu et al. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*, pages 21–37. Springer, 2016.

[10] Benjamin Planche and Eliot Andres. *Hands-On Computer Vision with TensorFlow 2*. Packt Publishing Ltd, 2019.

[11] Joseph Redmon et al. You only look once: Unified, real-time object detection. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 779–788, 2016.

[12] Joseph Redmon and Ali Farhadi. YOLO9000: Better, faster, stronger. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7263–7271, 2017.

[13] Joseph Redmon and Ali Farhadi. YoloV3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[14] Shaoqing Ren et al. Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 28:91–99, 2015.

[15] Roman Solovyev et al. Weighted boxes fusion: Ensembling boxes from different object detection models. *Image and Vision Computing*, 107:104117, 2021.

[16] Xu-Sheng Tang et al. Bagging-adaboost ensemble with genetic algorithm post optimization for object detection. In *2009 5th Int Conf. on Natural Computation*, volume 4, pages 528–534, 2009.

[17] Bo Yang et al. Lessons learned from accident of autonomous vehicle testing: An edge learning-aided offloading framework. *IEEE Wireless Communications Letters*, 9(8):1182–1186, 2020.

[18] Zhong-Qiu Zhao et al. Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems*, 30(11):3212–3232, 2019.

[19] Zhengxia Zou et al. Object detection in 20 years: A survey. *CoRR*, abs/1905.05055, 2019.