Adaptive Autonomy in Human-on-the-Loop Vision-Based Robotics Systems

Sophia Abraham*, Zachariah Carmichael*, Sreya Banerjee*, Rosaura VidalMata*,
Ankit Agrawal†, Md Nafee Al Islam†, Walter Scheirer*, Jane Cleland-Huang†
*Computer Vision Research Lab
†DroneResponse Lab
Department of Computer Science and Engineering
University of Notre Dame
South Bend, USA
{sabraha2, zcarmich, Sreya.Banerjee.9, rvidalma, aagrawa2, mislam2, wscheire, JaneHuang}@nd.edu

***Abstract*—Computer vision approaches are widely used by autonomous robotic systems to sense the world around them and to guide their decision making as they perform diverse tasks such as collision avoidance, search and rescue, and object manipulation. High accuracy is critical, particularly for Human-on-the-loop (HoTL) systems where decisions are made autonomously by the system, and humans play only a supervisory role. Failures of the vision model can lead to erroneous decisions with potentially life or death consequences. In this paper, we propose a solution based upon adaptive autonomy levels, whereby the system detects loss of reliability of these models and responds by temporarily lowering its own autonomy levels and increasing engagement of the human in the decision-making process. Our solution is applicable for vision-based tasks in which humans have time to react and provide guidance. When implemented, our approach would estimate the reliability of the vision task by considering uncertainty in its model, and by performing covariate analysis to determine when the current operating environment is ill-matched to the model's training data. We provide examples from DroneResponse, in which small Unmanned Aerial Systems are deployed for Emergency Response missions, and show how the vision model's reliability would be used in addition to confidence scores to drive and specify the behavior and adaptation of the system's autonomy. This workshop paper outlines our proposed approach and describes open challenges at the intersection of Computer Vision and Software Engineering for the safe and reliable deployment of vision models in the decision making of autonomous systems.**

***Index Terms*—computer vision, adaptive autonomy, safety, uncertainty**

I. Introduction

Computer Vision (CV) models are broadly utilized within autonomous systems. Examples include driving systems, factory-floor robots, and small Unmanned Aerial Systems (sUAS) deployed for emergency response missions. CV is essential to these applications as it provides critical information about the environment in which the system is operating, and this information is used to support autonomous decision-making. However, CV systems are not entirely reliable and can incorrectly identify, or fail to identify, objects in the real world, leading to incorrect autonomous decisions. One notable example of CV failure is Uber's Self-Driving car accident on March 19, 2018, which resulted in the fatality of a woman whilst the vehicle was running in autonomous mode with a human driver as a backup. The recorded telemetry showed the vision system had detected and classified the woman six seconds before the crash as an *unknown object*, then as a *vehicle*, and finally as a *bicycle*, resulting in varied predictive responses based on the car's inbuilt autonomy logic. The system finally recognized the need for emergency breaking 1.3 seconds prior to the impact, but that was too late. Uber stated that emergency braking maneuvers were not enabled in these circumstances to reduce "erratic vehicle behavior", and furthermore, that the system was not designed to alert the operator. A post-mortem analysis identified contributing causes as dark clothing on the pedestrian, lack of side reflectors on the bicycle, front/rear reflectors perpendicular to the path of the vehicle, and no roadway lighting at the location of the incident [1].

These types of CV failures have multiple root causes, many of which are introduced whilst training the CV models. For example, data bias may be introduced by imbalanced data, as in the Uber case, or as human-introduced bias (such as racial biases reflected in recidivism data, or gender biases reflected in census incomes [2]). Bias can cause a model to under-perform in certain circumstances. For example, Wilson et al, [3] reported that state-of-the-art object detection systems return poorer performance when detecting pedestrians with darker skin tones regardless of the time of day or whether the person is occluded. When studying the data used to train such models, researchers found that there were about 3.5 times as many samples of people with lighter skin tones than those of people with darker skin. As these kinds of problems are prevalent across almost all current CV models, software intensive systems that leverage CV models must be developed defensively in order to mitigate CV-induced risks.

Human-on-the-loop (HoTL) systems are empowered to make and enact their own decisions [4] with humans performing only a supervisory role [5]. Decisions are supported by the system's knowledge of the environment, which is often acquired using CV. In many CV-based scenarios it is essential for the human supervisor to understand whether the vision

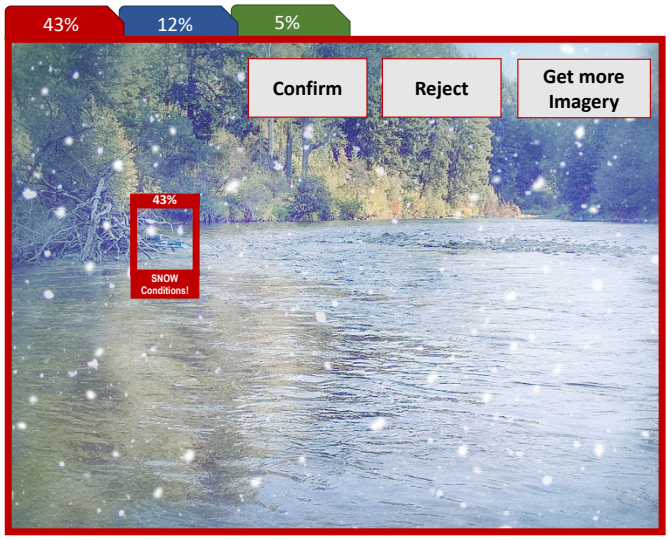

Fig. 1. The sUAS detects a victim with a medium level of confidence and low certainty. It requests confirmation from the human operator before raising a victim-found alert and aborting its search by switching to 'tracking' mode. The human can confirm, reject, or request additional imagery. One reason for low confidence is that the training data lacks snowy weather examples.

model can provide reliable results for a specific vision task within the current environment, and whether the system should be trusted to react autonomously, or whether human input is required. Consider the example shown in Figure 1 in which an sUAS has detected a potential victim, streams video, and requests input from the human operator. The system can be designed with varying degrees of autonomy with respect to how the sUAS reacts after detecting a candidate victim. For simplicity's sake in this discussion, we assume that only one sUAS has detected the candidate victim – in this case with a confidence score of 0.43. At this point, the sUAS can (1) ask the operator what to do, (2) automatically track the victim if its confidence exceeds a predefined fixed threshold, and then notify the operator of its actions (HoTL), or (3) decide whether it is able to start tracking based on the perceived reliability of the CV model within the current environment. The third option represents *adaptive autonomy* in which the sUAS operates independently when it trusts the information provided by the underlying CV models, and engages the operator in critical decisions when the model becomes unreliable.

The aim of this paper is to explore situations in which CV models may suffer from low reliability, propose techniques for detecting failures, and specify requirements for dynamically adjusting autonomy levels and triggering human intervention when the CV model is unable to perform reliably as illustrated in Fig. 2. We focus upon scenarios in which humans have time to react and to provide guidance if needed; however, our approach can also be used to raise alerts in situations where the system is making real-time decisions even though CV reliability is low. We leave the full implementation of the approach to future work. We draw examples from our own HoTL DroneResponse system which deploys multiple sUAS to support time-critical, emergency response missions [6], [7].

The remainder of the paper is structured as follows. Section II presents related work and lays the foundation for our discussion. Section III describes our proposed solution for assessing model reliability through evaluating uncertainty and performing covariate analysis with respect to the current environment. Section IV describes our software engineering solution for CV-driven decision making and autonomy adaptation, while Section V closes with a discussion of open research challenges, and Section VI closes with conclusions.

II. Background Information

A self-adaptive system is capable of reconfiguring at run-time in response to changes in the system and its environment [8]. Adaptations include changes in run-time behavior, often realized through switching modes of operation, or by reconfiguring parameters within a mode. For example, in our DroneResponse system, an sUAS switches between modes (e.g., Takeoff, Search, Track) in response to external events (e.g., destination reached), and can reconfigure its behavior within a mode (e.g., by changing altitude, or increasing monitoring frequency). However, in this paper, we focus on a special form of adaptation that occurs as a result of uncertainty in the CV model. This form of adaptation represents a temporary switch from HoTL behavior to HiTL (human-in-the-loop) behavior, when the CV model is perceived as unreliable for performing the current task. Many Software Engineering researchers have explored the notion of uncertainty and its role in self-adaptation [9] including work on identifying gaps in the training data that might make a CV model less reliable in certain environments [10]. AI systems, and CV ones in particular, can cause erratic behavior if they fail to produce accurate results [11]–[14].

Model Explainability: Much related work has focused on explaining predictions made by AI models, especially those leveraging deep neural networks, which tend to provide little or no human-readable rationale for their internal decisions. Their black-box nature can conceal biases, deficiencies, and dubious correlations, which are especially likely when "dataset shift" occurs between the training data and the current image stream [15]. Explainable AI (XAI) techniques add a layer of transparency to this process, whether it is a heat map of pixel importance [16], [17], auxiliary model (an explanation model built for each image) [18], [19], or using attribution based confidence metrics [20]. Such methods provide explanations in the form of feature attributions (how much they influence the magnitude of a prediction), similar examples, counterfactuals (examples of changes to an image that would have caused a different decision), rules, and visualizations, thus, introducing further clarity during the model development and deployment phases. For instance, feature attribution can indicate that a model relies too heavily on backgrounds of the images rather than the objects of interest in the foreground. Explainable AI can provide useful insights for testing and improving CV models. For example, it might be found that a CV model fails to detect target objects (e.g., people) in rainy conditions due

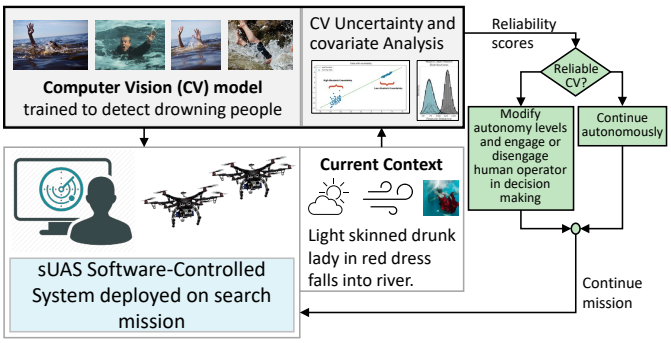

Fig. 2. The vision model is trained on a specific dataset which can introduce bias into the vision-based decision-making. This introduces the critical question of whether the model can be trusted to make a correct decision in the given context.

to partial occlusion of the object. This in turn could lead to modified training data including more rainy scenarios. However, explanations of CV decisions deployed in autonomous systems have limited runtime utility, when agents (e.g., sUAS) must make immediate adaptation decisions.

Uncertainty Estimation: Our approach requires the system to evaluate the reliability of the CV model and adapt autonomous behavior accordingly. Uncertainty can be quantified by considering the confidence with respect to the model, the data, and the physical sensor(s). Many approaches have been explored, including augmenting AI systems with auxiliary confidence-estimating modules [21], directly calibrating the decision probabilities [22], and taking Bayesian estimates of certainty [23]. The first two techniques involve training a separate component of the system to calibrate decisions as probabilities directly interpretable as confidence scores. The latter adds a supplemental model that estimates the conditional distribution of the model decisions given the system and its data. All techniques output the probability that the system will be reliable, regardless of whether the unreliability is caused by unfamiliar data, noise, or other confounding signals.

Integrating Humans in the Decision-Making Process: Finally, researchers have investigated when and where humans should be involved in decision-making processes in order to maximize the benefits of autonomy whilst interjecting human feedback when needed for correctness, safety, regulatory compliance, or other purposes. Examples include runtime supervision of autonomous driving systems for safety purposes [24], training machine learning solutions in health informatics to increase their accuracy [25], and human-robot partnerships in machine assembly tasks [26]. For example, Cai et al., [27] proposed an approach in which a robot collects multiple images of an object, classifies them using a trained deep convolutional neural network, and when confidence is low for some subset of the images, it determines whether it should autonomously reposition itself to potentially achieve a better viewpoint or whether it should request help from a remote operator. This approach is similar to our proposed adaptive autonomy solution.

III. CV-Supported Autonomy

DroneResponse deploys multiple sUAS to support emergency response missions [6], [7], [28]. It is designed as a HoTL system empowered to make and enact decisions [4] supported by its onboard CV. Enabling higher degrees of autonomy is particularly important in DroneResponse missions, where multiple sUAS are deployed simultaneously to perform time-constrained search-and-rescue, surveillance, or delivery tasks.

A. Estimating Loss of Reliability

Our approach determines the trust that the sUAS system should place in CV predictions based upon confidence and reliability of the underlying model, where confidence is defined as the probability that a CV decision is correct given the evidence it considers, and reliability is estimated using notions of *uncertainty* arising from noise and model or observation incompleteness [29]. Finally, covariate shift analysis is performed to determine the model's performance under multiple continuous covariates to allow us to estimate the model's reliability in any real-time operating condition. These covariates can be specified as any attributes that may affect the efficacy of the vision model within the given operating domain, such as the attributes depicted in Fig. 5 for the DroneResponse river-rescue scenario.

B. Modeling and Detecting Unreliability

Our approach utilizes two different techniques for estimating the loss of reliability. The first, is based on the formal notion of uncertainty, derived using a Bayesian Belief Network (BBN), while the second is based on an estimation of the covariate shift between the current scene and the data used to train the model. We discuss each of these approaches.

Uncertainty in CV Models: Uncertainty is typically estimated using Bayesian surrogate estimators, that enable the CV algorithm to infer its own degree of certainty, represented as one or more probability scores. We adopt state-of-the-art techniques in uncertainty estimation. Loquercio et al., [30] proposed the use of BBNs and Monte-Carlo sampling to derive uncertainty from both the data and model. Specifically, data noise, arising from the sensor, is assumed to follow a normal distribution based on known noise characteristics. In our sUAS system, these characteristics could include glare, excess vibration caused by the sUAS and/or the camera mount, physical occlusion of objects to be detected, or general image noise. Uncertainty among the parameters is estimated by Monte Carlo sampling of the parameters with test time dropout – that is, reducing the population before sampling for the sake of tractability. The model makes several predictions using such subsets, the variance of which is the model certainty. The full system uncertainty is then the total variance of the data uncertainty propagated through the model. In other words, it is the the extent to which the estimation of uncertainty changes between each layer of the CV model, from that induced by sensor noise up to the model parameters of the output layer.

We integrate this technique as a module into our proposed framework, providing a streaming confidence interval in the range $[0, 1]$, which comprises uncertainty estimates from both the camera and the vision model, accounting for noise and incompleteness. As a scaffold, we add a hysteresis band that filters transient noise, making the algorithm less sensitive to minor fluctuations which would otherwise produce false positives or negatives. The video stream is transformed into semantic data representing *confident* (no intervention), *uncertain* (possible intervention), and *no confidence* (intervention required). These parameters can be tuned using field data and adjusted according to intervention budgets. In the case of DroneResponse they would be adjusted based upon a combination of safety factors, event occurrence, and human resources. For example, if multiple sUAS raised alerts at the same time, and the human operator is unable to process all of them, then the events must be prioritized for intervention, while other sUAS make their best judgments until the operator is able to review, and then confirm or refute their decisions.

Covariate Shift Analysis: In addition to estimating uncertainty generated by the model, it can be helpful to understand the effects of covariates on the model's performance within an operating context. Covariates refer to the known measured attributes within the data. A situation known as *covariate shift* occurs when the training data differs from the data seen at the time a prediction is made. This can result in incorrect predictions with high levels of confidence [31], a problem which is common in real-time detection models. For example, a CV model trained only in good visibility conditions, might underperform in low visibility.

We propose the formation of a generative model based on the work of McCurrie et al. [32] to capture the distribution between multiple continuous covariates and model performance and to subsequently inform the decision making process.

In reference to a person detection model, the generative model could be formulated in three distinct steps:

1) *Data annotation:* Given a dataset containing N images, we annotate the images with relevant covariates. While covariates such as age, gender, and race might need to be manually annotated, other covariates such as clothing, angle of view, occlusion, environmental factors, and weather conditions could be extracted with attribute classifiers. Two classifiers that we have developed include a weather classifier and a semantic segmentation model for labeling different parts of the river. Our weather model is built using an ensemble of binary support vector machines trained for pertinent attributes (i.e., light, rain, snow, etc.). It tags each of the N images with corresponding weather and daylight meta-characteristics (Figure 3). Our semantic segmentation model tags data with setting and terrain meta-characteristics (e.g., water, river bank). For this segmentation model, we utilized DeepLab [33] for relevant operating covariates (Figure 4).
2) *Estimate pairwise similarities:* The pairwise distance or similarity between the dataset and the image itself is calculated by constructing a matrix in with N^2 data points where each row (query) represents the image being sampled, and each column (gallery) represents the collection of images on which the generated model has been formed. Given a data point (X_k, y_k) for $k \in \{1, ..., N^2\}$, y_k represents the similarity between two images and X_k is the vector of all the query and gallery attributes. This also includes a user-defined Boolean attribute which returns 'true' when the query and gallery attributes are matched and 'false' when they are not matched.
3) *Density regression:* Finally, we estimate the full density of the general distribution of predictive scores over the continuous covariates in order to determine the extent to which the current image from the operating context matches the images in the training set. We first calculate the conditional true positive rate (TPR):

$$TPR(fpr|X) = F_M(F_{\bar{M}}^{-1}(fpr|X)|X)$$

This provides the TPR at a given false positive rate (fpr) for a precondition X, which is the covariate vector mentioned before. Here F_M and $F_{\bar{M}}$ are the cumulative distribution functions (CDF) of the match and non-match distribution. The above relation can be utilized at runtime to estimate the reliability of the model given the covariates of the operating condition.

By extracting and analyzing covariates in this way, and by detecting uncertainty in the CV models with respect to the current CV-task, the system can determine the extent to

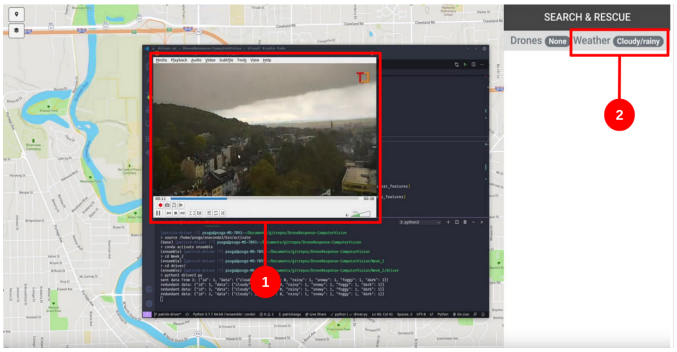

Fig. 3. Our trained weather classifier can take a video stream (1) and tag with weather characteristics (2). Covariate shift can be assessed by comparing the weather distribution for the training data against the current image stream.

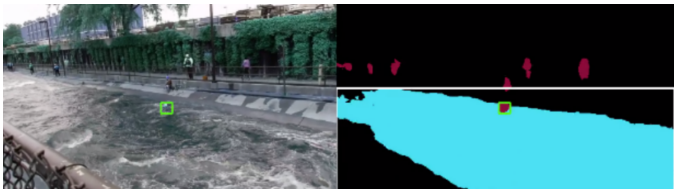

Fig. 4. Our use of the DeepLab [33] semantic segmentation model applied to video acquired for a river rescue scenario from the South Bend Fire Department, successfully encoded the pixels corresponding to 'river' and 'people' from the frame of reference (right panel). Segmentation models could be used to automate the meta-tagging of pertinent covariates.

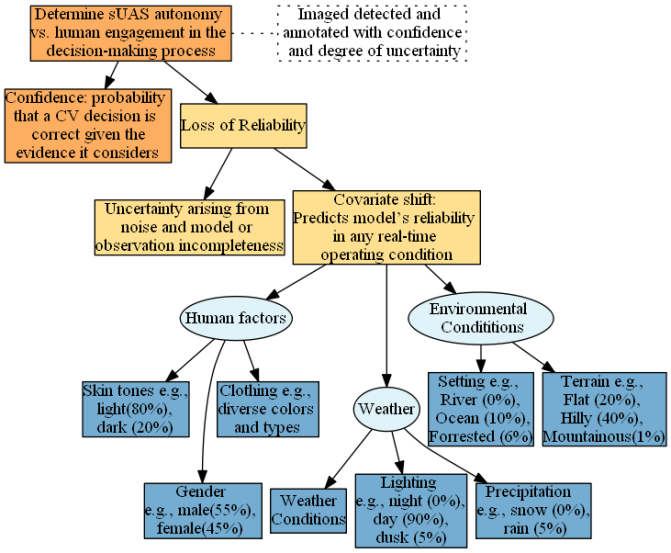

Fig. 5. The model's Meta-characteristics are identified for the domain of river search and rescue, that could impact the trustworthiness of the Vision Model. Numbers in the leaf nodes depict the percentage of training samples from the dataset that match each of the meta-tags. Note: These numbers are not derived from actual datasets and are used for illustrative purposes only.

which it can make autonomous decisions and whether human intervention is required.

IV. Specifying Autonomy Adaptation Requirements

Our framework supports the development of a CV-driven solution with awareness of the model's reliability. Given this self-awareness, we can specify requirements and design a system capable of adapting its own autonomous behavior according to the perceived reliability. Whittle et al., [9] previously proposed the RELAX language for specifying requirements of self-adaptive systems. RELAX provides the means of describing uncertainty using natural language or Fuzzy branching temporal logic and supports the notion of requirements satisficing in which requirements can be relaxed to address uncertainty. However, in this initial paper, we adopt the simpler EARS (Easy Requirements Specification) notation [34], [35] which is sufficiently expressive to define adaptive behaviors proposed by our approach.

We identify human-sUAS interaction points by leveraging our existing meta-model for human-on-the-loop interactions in multi-agent missions [7]. In addition to modeling typical human-sUAS interactions, the model includes 'probing questions' designed to aid in the discovery and specification of autonomy requirements. The following three questions are particularly pertinent to our discussion of CV-related autonomy and human interventions.

Q1: When and where do the agents exhibit **autonomous decision-making behavior?**

Q2: Under **normal operating conditions**, what decisions should the agent be able to make autonomously?

Q3: How is the **autonomy suppressed or increased** at this interaction point? (e.g., modifying the confidence threshold for automatically tracking a potential victim, disabling/enabling the ability to track without permission, ...)

We answer these questions in order to specify system level requirements which ultimately must be realized through lower level software requirements and design constraints. Here we focus primarily on design-level requirements which specify rules for switching between autonomous behavior and human-intervention. We err on the side of caution and engage the operator in the decision making process whenever (1) loss of reliability of the CV model exceeds a predefined threshold and/or (2) when the covariate shift indicates that the CV model training data does not provide sufficient coverage for the current environment. We establish internal thresholds for each of these and define *loss_of_reliability=TRUE* when either uncertainty exceeds *uncertainty_threshold* or the covariate analysis returns a score $<$ *covariate_coverage*.

We apply the questions (Q1, Q2, Q3) to aid in specifying the requirements for autonomy adaptation of our running example (see Figure 1) in which the sUAS detects a victim and determines what actions should be taken (Q1). To answer Q2 we specify the following autonomy requirements (AR) where CS is the confidence score generated by the CV model.

AR1: When the CV model identifies a candidate victim in the river with $CS >=$ *detect_threshold* and *loss_of_reliability=FALSE* then the sUAS autonomously transitions into *tracking* mode and notifies the operator.

AR2: When the CV model identifies a candidate victim in the river with $CS >=$ *detect_threshold* and *loss_of_reliability=TRUE* then the sUAS temporarily reduces its autonomy level, and raises a high-priority alert requesting permission from the operator to transition into *tracking* mode.

AR3: When the CV model identifies a candidate victim in the river with *alert_threshold* $<$ CS $<$ *detect_threshold* then the sUAS raises a *low-priority alert* and continues with its current tracking task.

We address Q3 at the design level. Instead of the system limiting the autonomy of the sUAS, we imbue the sUAS with self-awareness so that it detects loss of reliability of the CV model and temporarily overrides its own autonomy to request help in its decision-making ability.

Returning to our earlier example in which an sUAS detects a candidate victim in the river (cf Fig. 1), if we assume that *detect_threshold* if 0.4, but *loss_of_reliability*='true' due to the presence of snow in the operating environment without sufficient representation in the training set, the requirement *AR2* is activated and the human is alerted about the potential victim sighting, and engaged in the decision-making process. In this case, the human might request 'get more imagery' triggering the sUAS to reposition itself and stream further imagery, or could reject the sighting and direct the sUAS to continue its search.

This workshop paper focuses on describing how CV confidence, uncertainty, and covariate shift can be used to specify autonomous behavior. Experimental analysis of thresholds, and full integration of our approach within DroneResponse is left for future work.

V. Open Challenges

This paper has laid out a practical approach for leveraging CV models within autonomous systems. It describes our proposed framework, which in turn builds upon cutting edge research from both the Computer Vision and Software Engineering communities. To deploy the proposed solution on sUAS operating in potentially unknown environments requires us to address a number of open challenges associated with the CV models themselves, and their safe and reliable adoption within autonomous systems.

Challenge #1, Discovering & assessing salient covariates: As discussed in section III-B, identifying covariates is a challenging problem due to the black box nature of neural networks and the vast space of potential covariates. We need the ability to determine which covariates to include in our reliability model, the extent to which they impact performance (individually and as a group) and ways to understand prominent covariates which may not be human-comprehensible. Some related work has been in coercing CV models to discover features in training and visualizing features, e.g. [36]. Activation maps can indicate what features invoke larger responses from a model and thus pave the way to covariate discovery. The challenge here is to identify the covariates that affect reliability, extract covariates at runtime during an sUAS deployment, and to estimate uncertainty based on these factors.

Challenge #2, Extracting attributes from image data: Most datasets used to train vision models are assembled to support specific tasks. For example, object detection and recognition datasets such as PASCAL VOC [37], Imagenet [38], and MS-COCO [39] consist of thousands of images with labels for common objects, such as "person" with thousands of images representing people. However, none of these particular datasets take into consideration latent characteristics within the image such as light conditions, weather, terrain, or diverse characteristics of the people themselves. As a result, models trained on these datasets can fail due to out-of-distribution inputs [40], [41]. For instance, object detectors such as YOLO [42] trained on the PASCAL dataset, may perform better when localizing people on a bright sunny day than on a rainy or a snowy day since the dataset does not have sufficient representation of images under such weather conditions. When deployed in HoTL environments, their failures can lead to erroneous decisions. However, manually annotating the training set with this information, in order to evaluate covariate shift is time-consuming and difficult. The challenge is therefore to develop, reliable and automated techniques for meta-tagging covariates in the training set. We provided examples of two techniques for detecting weather conditions and identifying elements of a scene using semantic segmentation.

Challenge #3, Capturing real-time context: In addition to understanding the distribution of covariates in the training data, we also need to detect relevant covariates within the operating context at runtime. Diverse information sources can be leveraged, such as weather services, onboard sensors, and automated meta-taggers – potentially using the same classifiers that were applied to the training data. In addition, human input can be elicited to take advantage of operators' perspectives of the environment.

Challenge #4: Safety Assurance of CV-Driven systems Prior research has developed techniques for addressing safety assurance for self-adaptive systems [43]–[45]; however, there is a need for closer integration of the three-way interplay between reliability of state of the art CV models and predictions, the adaptive role of human engagement, and the subsequent creation and generation of safety assurance cases (SAC) [46]–[49] which provide evidence for system safety [50]–[52]. In particular, the safety case must show that thresholds are set at levels that effectively balance agent autonomy with human intervention.

Challenge #5: Human-Machine Interaction Our proposed approach engages humans-in-the-loop with the aim of increasing accuracy of the CV-driven decision making. However, prior work has identified different hazards that are introduced when humans place undue trust in the behavior of an autonomous system and as a result, fall out-of-the-loop [53], and engage in decision making without sufficient situational awareness. The engagement of humans in supervising and intervening in CV tasks requires careful analysis to explore the tradeoffs of introducing new Human-Computer interactions errors.

VI. Conclusions

This paper has presented an adaptive, informal framework for supporting the reliable deployment of CV models in the decision making of autonomous systems and the associated open challenges. It has proposed a framework for determining the reliability of the CV model by estimating the uncertainty of the model and capturing the covariate shift. The level of autonomy of the system is determined according to the model's reliability. In addition, the paper has described an approach for identifying and specifying uncertainty-driven autonomy requirements driven. We have presented proof-of-concept techniques for parts of our proposed solution, but have not yet integrated and fully validated all the pieces within our DroneResponse system.

Acknowledgment

This work was partially funded by the US National Science Foundation under grant CNS:1931962. We thank undergraduate students Soumya Abraham and Patrick Soga and graduate student Eric Tsai for helping to develop the weather detection models. We further thank undergraduate students Mike Prieto, Luke Siela, Ben Merrick, Bridget Hart, and Joseph

DelleDonne for developing the UI and streaming capabilities to support onboard vision in DroneResponse.

REFERENCES

[1] "Preliminary report highway hwy18mh010," 2018. [Online]. Available: https://www.ntsb.gov/investigations/AccidentReports/Reports/HWY18MH010-prelim.pdf

[2] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[3] B. Wilson, J. Hoffman, and J. Morgenstern, "Predictive inequity in object detection," *arXiv preprint arXiv:1902.11097*, 2019.

[4] J. E. Fischer, C. Greenhalgh, W. Jiang, S. D. Ramchurn, F. Wu, and T. Rodden, "In-the-loop or on-the-loop? interactional arrangements to support team coordination with a planning agent," *Concurrency and Computation: Practice and Experience*, p. e4082, 2017.

[5] S. Nahavandi, "Trusted autonomy between humans and robots: Toward human-on-the-loop in robotics and autonomous systems," *IEEE Systems, Man, and Cybernetics Magazine*, vol. 3, no. 1, pp. 10–17, 2017.

[6] A. Agrawal, S. J. Abraham, B. Burger, C. Christine, L. Fraser, J. M. Hoeksema, S. Hwang, E. Travnik, S. Kumar, W. J. Scheirer, J. Cleland-Huang, M. Vierhauser, R. Bauer, and S. Cox, "The next generation of human-drone partnerships: Co-designing an emergency response system," in *CHI*. ACM, 2020, pp. 1–13.

[7] A. Agrawal, J. Cleland-Huang, and J. Steghöfer, "Model-driven requirements for humans-on-the-loop multi-uav missions," in *MoDRE@RE*. IEEE, 2020, pp. 1–10.

[8] E. M. Fredericks and B. H. C. Cheng, "An empirical analysis of providing assurance for self-adaptive systems at different levels of abstraction in the face of uncertainty," in *SBST@ICSE*. IEEE Computer Society, 2015, pp. 8–14.

[9] J. Whittle, P. Sawyer, N. Bencomo, B. H. C. Cheng, and J. Bruel, "Relax: Incorporating uncertainty into the specification of self-adaptive systems," in *2009 17th IEEE International Requirements Engineering Conference*, 2009, pp. 79–88.

[10] M. A. Langford and B. H. C. Cheng, "Enhancing learning-enabled software systems to address environmental uncertainty," in *2019 IEEE International Conference on Autonomic Computing (ICAC)*, 2019, pp. 115–124.

[11] R. Yampolskiy, "Incident number 48," *AI Incident Database*, 2016.

[12] R. Yampolskiy and C. Pownall, "Incident number 60," *AI Incident Database*, 2017.

[13] J. Buolamwini and T. Gebru, "Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification," in *Conference on Fairness, Accountability and Transparency*. PMLR, Jan. 2018, pp. 77–91.

[14] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann, "Shortcut learning in deep neural networks," *Nature Machine Intelligence*, vol. 2, no. 11, pp. 665–673, Nov. 2020.

[15] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset Shift in Machine Learning*. The MIT Press, 2009.

[16] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation," *PLOS ONE*, vol. 10, no. 7, p. e0130140, Jul. 2015.

[17] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017, pp. 618–626.

[18] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 4765–4774.

[19] M. T. Ribeiro, S. Singh, and C. Guestrin, ""Why Should I Trust You?": Explaining the Predictions of Any Classifier," in *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: ACM, 2016, pp. 1135–1144.

[20] S. Jha, S. Raj, S. Fernandes, S. K. Jha, S. Jha, B. Jalaian, G. Verma, and A. Swami, "Attribution-based confidence metric for deep neural networks," in *Advances in Neural Information Processing Systems*, 2019, pp. 11 826–11 837.

[21] C. Corbière, N. Thome, A. Bar-Hen, M. Cord, and P. Pérez, "Addressing failure prediction by learning model confidence," in *Advances in Neural Information Processing Systems*, 2019, pp. 2902–2913.

[22] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 2017, pp. 1321–1330. [Online]. Available: http://proceedings.mlr.press/v70/guo17a.html

[23] A. Malinin and M. Gales, "Predictive uncertainty estimation via prior networks," in *Advances in Neural Information Processing Systems*, 2018, pp. 7047–7058.

[24] M. Gil, M. Albert, J. Fons, and V. Pelechano, "Designing human-in-the-loop autonomous cyber-physical systems," *International Journal of Human-Computer Studies*, vol. 130, pp. 21 – 39, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1071581919300461

[25] A. Holzinger, "Interactive machine learning for health informatics: when do we need the human-in-the-loop?" *Brain Informatics*, vol. 3, no. 2, pp. 119–131, Jun 2016. [Online]. Available: https://doi.org/10.1007/s40708-016-0042-6

[26] M. Raessa, J. C. Y. Chen, W. Wan, and K. Harada, "Human-in-the-loop robotic manipulation planning for collaborative assembly," 2019.

[27] H. Cai and Y. Mostofi, "Exploiting object similarity for robotic visual recognition," *IEEE Transactions on Robotics*, pp. 1–18, 2020.

[28] A. Agrawal, J. Cleland-Huang, and J.-P. Stegh"ofer, "Model-driven requirements for humans-on-the-loop multi-uav missions," in *2020 IEEE Tenth International Model-Driven Requirements Engineering (MoDRE)*. IEEE, 2020, pp. 1–10.

[29] A. Pouget, J. Drugowitsch, and A. Kepecs, "Confidence and certainty: distinct probabilistic quantities for different goals," *Nature neuroscience*, vol. 19, no. 3, p. 366, 2016.

[30] A. Loquercio, M. Segu, and D. Scaramuzza, "A general framework for uncertainty estimation in deep learning," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3153–3160, 2020.

[31] Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. Dillon, B. Lakshminarayanan, and J. Snoek, "Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019, pp. 13 991–14 002. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/8558cb408c1d76621371888657d2eb1d-Paper.pdf

[32] M. McCurrie, H. Nicholson, W. J. Scheirer, and S. Anthony, "Modeling score distributions and continuous covariates: A bayesian approach," in *2020 IEEE International Joint Conference on Biometrics (IJCB)*, 2020, pp. 1–9.

[33] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[34] A. Mavin, P. Wilkinson, A. Harwood, and M. Novak, "Easy approach to requirements syntax (ears)," in *2009 17th IEEE International Requirements Engineering Conference*, 2009, pp. 317–322.

[35] J. Cleland-Huang and M. Vierhauser, "Discovering, analyzing, and managing safety stories in agile projects," in *RE*. IEEE Computer Society, 2018, pp. 262–273.

[36] W. Yang, H. Huang, Z. Zhang, X. Chen, K. Huang, and S. Zhang, "Towards rich feature discovery with class activation maps augmentation for person re-identification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[37] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *IJCV*, vol. 88, no. 2, pp. 303–338, 2010.

[38] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "ImageNet large scale visual recognition challenge," *IJCV*, vol. 115, no. 3, pp. 211–252, 2015.

[39] 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.

[40] R. G. VidalMata, S. Banerjee, B. RichardWebster, M. Albright, P. Davalos, S. McCloskey, B. Miller, A. Tambo, S. Ghosh, S. Nagesh *et al.*, "Bridging the gap between computational photography and visual recognition," *arXiv preprint arXiv:1901.09482*, 2019.

[41] S. Banerjee, R. G. VidalMata, Z. Wang, and W. J. Scheirer, "Report on ugˆ 2+ challenge track 1: Assessing algorithms to improve video object detection and classification from unconstrained mobility platforms," *arXiv preprint arXiv:1907.11529*, 2019.

[42] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[43] B. H. C. Cheng, R. J. Clark, J. E. Fleck, M. A. Langford, and P. K. McKinley, "AC-ROS: assurance case driven adaptation for the robot operating system," in *MoDELS*. ACM, 2020, pp. 102–113.

[44] S. Jahan, M. Pasco, R. F. Gamble, P. K. McKinley, and B. H. C. Cheng, "MAPE-SAC: A framework to dynamically manage security assurance cases," in *FAS*W@SASO/ICAC*. IEEE, 2019, pp. 146–151.

[45] M. Trapp and D. Schneider, "Safety assurance of open adaptive systems - A survey," in *Models@run.time@Dagstuhl*, ser. Lecture Notes in Computer Science, vol. 8378. Springer, 2011, pp. 279–318.

[46] R. Bloomfield and P. Bishop, "Safety and assurance cases: Past, present and possible future–an Adelard perspective," in *Making Systems Safer*. Springer, 2010, pp. 51–67.

[47] C. M. Holloway, "Safety case notations: Alternatives for the non-graphically inclined?" in *Proc. of the 3rd IET Int'l Conf. on System Safety*. IET, 2008, pp. 1–6.

[48] R. Hawkins, I. Habli, T. Kelly, and J. McDermid, "Assurance cases and prescriptive software safety certification: A comparative study," *Saf. Sci.*, vol. 59, pp. 55–71, 2013.

[49] T. Kelly and R. Weaver, "The Goal Structuring Notation–a safety argument notation," in *Proc. of the Dependable Systems and Networks 2004 WS on Assurance Cases*. Citeseer, 2004, p. 6.

[50] J. Chen, M. Goodrum, R. A. Metoyer, and J. Cleland-Huang, "How do practitioners perceive assurance cases in safety-critical software systems?" in *CHASE@ICSE*. ACM, 2018, pp. 57–60.

[51] U.K. Ministry of Defence, "Defence Standard 00-56, Issue 7: Safety Management Requirements for Defence Systems. Part 1: Requirements," 2017.

[52] P. J. Graydon and C. M. Holloway, "An investigation of proposed techniques for quantifying confidence in assurance arguments," *Saf. Sci.*, vol. 92, pp. 53–65, feb 2017.

[53] M. R. Endsley, *Designing for Situation Awareness: An Approach to User-Centered Design, Second Edition*, 2nd ed. Boca Raton, FL, USA: CRC Press, Inc., 2011.