StreamCollab: A Streaming Crowd-AI Collaborative System to Smart Urban Infrastructure Monitoring in Social Sensing

Yang Zhang[1], **Lanyu Shang**[2], **Ruohan Zong**[1], **Zeng Wang**[1], **Ziyi Kou**[2], **Dong Wang**[2]

[1] Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN, USA
[2] School of Information Sciences, University of Illinois Urbana-Champaign, Champaign, IL, USA
yzhang42@nd.edu, lshang3@illinois.edu, rzong@nd.edu, zwang37@nd.edu, ziyikou2@illinois.edu, dwang24@illinois.edu

Abstract

Social sensing has emerged as a pervasive and scalable sensing paradigm to collect observations of the physical world from human sensors. A key advantage of social sensing is its infrastructure-free nature. In this paper, we focus on a *streaming urban infrastructure monitoring (Streaming UIM)* problem in social sensing. The goal is to automatically detect the urban infrastructure damages from the streaming imagery data posted on social media by exploring the collective power of both AI and human intelligence from crowdsourcing systems. Our work is motivated by the limitation of current AI and crowdsourcing solutions that either fail in many critical time-sensitive UIM application scenarios or are not easily generalizable to monitor the damage of different types of urban infrastructures. We identify two critical challenges in solving our problem: i) it is difficult to dynamically integrate AI and crowd intelligence to effectively identify and fix the failure cases of AI solutions; ii) it is non-trivial to obtain accurate human intelligence from unreliable crowd workers in streaming UIM applications. In this paper, we propose *StreamCollab*, a streaming crowd-AI collaborative system that explores the collaborative intelligence from AI and crowd to solve the streaming UIM problem. The evaluation results on a real-world urban infrastructure imagery dataset collected from social media demonstrate that StreamCollab consistently outperforms both state-of-the-art AI and crowd-AI baselines in UIM accuracy while maintaining the lowest computational cost.

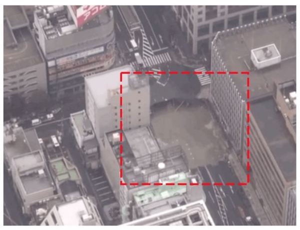
Introduction

Social sensing has emerged as a pervasive and scalable sensing paradigm to collect observations of the physical world from human sensors (Wang et al. 2019a, 2012). Examples of social sensing applications include urban air quality assessment with reports from citizen scientists (Dutta et al. 2016), real-time traffic condition monitoring using crowdsensing data (Lin and Li 2020), and disaster response with social media posts (Wang et al. 2014). A key advantage of social sensing is its infrastructure-free nature that does not require any infrastructure sensors (e.g., surveillance cameras, radar sensors) in the data collection process, making such sensing paradigm pervasive and scalable (Wang, Abdelzaher, and Kaplan 2015). In this paper, we study an emerging problem – *streaming urban infrastructure monitoring (Streaming UIM)* (Figueiras et al. 2018) – in social sensing. The goal is to automatically detect the urban infrastructure damages from the streaming imagery data posted on social media. The outputs of the streaming UIM applications can be used by local and federal governments and infrastructure management authorities (e.g., Department of Public Works, Department of Construction) to provide timely repair and maintenance actions to the sites and save them from further damages and severe consequences (Zhao et al. 2018).

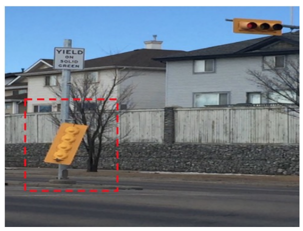
Recent advances in artificial intelligence (AI) and crowdsourcing have been applied to address the UIM problem (Oshri et al. 2018; Kankanamge et al. 2019; Harris et al. 2017; Alavi and Buttlar 2019). On one hand, the AI-based UIM solutions significantly improve the computational efficiency in streaming UIM applications compared to the traditional UIM solutions that are mainly done by the domain specialists (Oshri et al. 2018; Kankanamge et al. 2019). However, we observe that current AI-based solutions could fail in many critical time-sensitive UIM application scenarios. Figure 1 shows a few examples of such failure scenarios of AI-based solutions. We observe that the AI solutions miss the critical urban infrastructure damages such as broken stop sign, cracked building, collapsed road, and fallen traffic light. Those urban infrastructure damages could lead to severe consequences if those damages are not detected and responded rapidly. For example, the broken stop sign in Figure 1(a) could lead to severe traffic accidents and the cracked building in Figure 1(b) could result in a building collapse. On the other hand, recent efforts in citizen science and crowdsourcing have been made to overcome the limitation of the AI solutions by incentivizing citizen workers to monitor and report the potential urban infrastructure damages in a timely manner (e.g., SeeClickFix, FixeMyStreet, ImproveMyCity) (Harris et al. 2017; Alavi and Buttlar 2019). Citizen workers are often observed to perform better at identifying the urban infrastructure damage where AI solutions fail (Mandel et al. 2020). However, the current crowdsourcing solutions are often designed for a specific UIM task (e.g., monitoring road conditions, detecting building damages) and require extensive efforts to develop, distribute, and maintain the crowdsourcing apps to support the UIM tasks (Seto and Sekimoto 2019). Additionally, the actual UIM performance of these efforts often largely depends on the penetration ratio of the apps, which is always a challenge (Wang et al. 2017). As a result, the current crowdsourcing solutions are not easily generalizable to monitor the damage of different types of urban infrastructures as shown in Figure 1.

(a) Broken Stop Sign

(b) Cracked Building



(c) Collapsed Road



(d) Fallen Traffic Light

Damaged Urban Infrastructure recognized as Non-Damaged by AI (The red rectangles indicate the actual damaged areas)

Figure 1: Examples of Failure Cases from AI-based in Streaming UIM Applications

To address the limitations of current solutions, this paper develops a social sensing-based crowd-AI collaborative system to solve the streaming UIM problem by leveraging the collaborative intelligence from both AI and the human intelligence from crowdsourcing systems. On one hand, our system is designed to leverage AI's high efficiency to examine the vast amount of streaming social sensing data to identify potential infrastructure damages. On the other hand, our system dynamically incorporates the human intelligence from crowdsourcing systems to identify and fix the failure cases of AI to boost overall UIM performance. To obtain the human intelligence, we leverage the well-known crowdsourcing platform (i.e., Amazon Mechanical Turk (AMT)[1]) to acquire timely crowd responses about potential infrastructure damages reported in social sensing data. We refer to the human intelligence from crowdsourcing systems as *crowd intelligence* in our paper. While the design of our solution allows it to explore the collective intelligence of both human and machine intelligence and be generalizable to detect different types of urban infrastructure damage (e.g., the ones shown in Figure 1), it is not a trivial task to develop such a streaming crowd-AI collaborative system due to two critical challenges that we elaborate below.

Streaming Crowd-AI Collaboration. The first challenge lies in how to *dynamically* integrate AI and crowd intelligence to effectively identify and fix the failure cases of AI solutions in streaming UIM applications. One straightforward solution to address this problem is to directly ask the crowd workers to examine every output of AI solutions to identify and fix the failure cases as shown in Figure 1. However, such an approach is impractical due to the heavy labor costs and low efficiency, which could lead to a significant delay in streaming UIM applications (Pan et al. 2016). Recent efforts in crowd-AI hybrid systems were made to address this issue by only selecting the imagery data with complicated image property for crowd labeling under the assumption that the AI solutions are more likely to fail when the image is complex (Zhang et al. 2019; Sener and Savarese 2018). Those approaches then use the collected crowd labels to retrain the AI models to capture the dynamics of the streaming data. However, the AI model retraining process could lead to a non-negligible delay in providing timely responses to the UIM applications. Therefore, it remains to be a challenging question on how to design a streaming crowd-AI collaboration system to ensure desirable streaming UIM performance.

Uncertainty in Streaming Crowd Intelligence. The second challenge refers to the fact that it is difficult to obtain accurate human intelligence from unreliable crowd workers in streaming UIM applications. Unlike urban infrastructure labels obtained from domain experts in infrastructure management, the crowd annotations are often noisy and inconsistent due to the intrinsic uncertainty of crowd workers (e.g., lack of professional knowledge in infrastructure or civil engineering, conflicting responses from different crowd workers) (Hansson and Ludwig 2019). As a result, the current crowd-AI models could encounter a non-trivial performance loss when those models are trained on the imperfect crowd labels. Recent efforts address this problem by leveraging the crowd responses from a large number of data samples to build an effective de-noising model to obtain reliable crowd responses (Li et al. 2016; Zhang et al. 2019). However, there exists a "cold start" problem in our streaming UIM application, where the crowd responses on a large number of social media data are often not available at the beginning of the application (Mavridis, Gross-Amblard, and Miklós 2016). Therefore, it remains to be a challenging task to derive the accurate crowd intelligence in streaming UIM applications.

To address the above challenges, we develop *StreamCollab*, a streaming crowd-AI collaborative system that explores the collaborative intelligence from AI and crowd to solve the streaming UIM problem. To address the first challenge, we develop a dynamic deep uncertainty-aware estimation network that quantifies the uncertainty of the infrastructure damage estimation results and uses the estimated uncertainty to effectively detect the failure cases of AI. To address the second challenge, we develop a streaming crowd knowledge fusion framework that updates the estimation of unknown infrastructure damage labels from crowd responses to fix the failure cases of AI through a novel recursive estimation model. To the best of our knowledge, our StreamCollab is the first *streaming* crowd-AI hybrid system to dynamically fuse the AI and crowd intelligence under a principled analytical framework to address

[1]https://www.mturk.com

the streaming UIM problem. We evaluate the StreamCollab using a real-world urban infrastructure imagery dataset collected from social media. The evaluation results show that StreamCollab consistently outperforms both state-of-the-art AI-approaches and crowd-AI baselines in correctly identifying the damaged urban infrastructure while maintaining the lowest computational cost under various types of evaluation scenarios.

Related Work

Social Sensing

Social sensing has emerged as a new and pervasive sensing paradigm where human sensors collectively report timely observations about the physical world (Wang et al. 2019a). Examples of social sensing applications include real-time traffic condition monitoring using social media data (Laubis et al. 2019), natural disaster and emergency response systems by leveraging user's posts from social media feeds (Zhang et al. 2016), tracking urban environment conditions with self-reports from citizen scientists and engineers (Huang et al. 2018), and detecting infectious disease outbreaks in urban areas using location-based crowd tracking services (Mejova, Weber, and Fernandez-Luque 2018). Several key challenges exist in social sensing applications, including data sparsity, data reliability, real-time response guarantee, and privacy perseverance (Capponi et al. 2019; Wang, Kaplan, and Abdelzaher 2014; Zhang et al. 2020a). However, it remains a critical challenge to dynamically integrate AI and crowd intelligence to solve the streaming UIM problem in social sensing applications. In this paper, we develop a novel streaming crowd-AI collaborative system to accurately detect urban infrastructure damages in streaming social media imagery data.

Urban Infrastructure Monitoring

Previous efforts in urban infrastructure monitoring have leveraged AI and crowdsourcing technique to detect and assess the infrastructure damage in urban areas (Harris et al. 2017; Zhang and Wang 2015; Mei and Gül 2019; Wang et al. 2019b; Cervone et al. 2017). For example, Harris *et al.* developed a crowdsourcing monitoring paradigm to assess transportation infrastructure conditions using multimedia data collected from volunteer citizens (Harris et al. 2017). Mei *et al.* proposed a bridge damage detection framework by analyzing mobile sensor data from vehicles traveling on bridges (Mei and Gül 2019). Wang *et al.* designed a mobile crowdsourcing based urban road crack detection system that identifies road crack damage by aggregating crowdsourced image data and mobile sensor data (Wang et al. 2019b). Cervone *et al.* incorporated social media feeds with satellite images to assess urban transportation damages during emergencies (Cervone et al. 2017). Cervone *et al.* incorporated social media feeds with satellite images for road damage assessment during disaster events (Cervone et al. 2017). However, those AI-based solutions could fail in many time-sensitive UIM application scenarios and lead to further damages and severe consequences. This is because the AI solutions often lack a clear understanding of different types of complex urban infrastructure damages in the presence of a vast amount of streaming social media data (Hand 2017). Meanwhile, recent efforts in citizen science and crowdsourcing overcome the limitation of current AI solutions by actively recruiting participants to examine the urban infrastructure conditions and report any potential infrastructure damages through crowdsourcing apps (e.g., SeeClickFix, FixeMyStreet, ImproveMyCity) (Alavi and Buttlar 2019; Mandel et al. 2020). However, those solutions are not easily generalizable to different types of UIM applications due to the extensive efforts required to develop and manage those crowdsourcing apps and the challenge of keeping a sufficient penetration ratio of the apps. In contrast, we develop a social sensing-based crowd-AI collaborative system that explicitly explores the rich urban infrastructure damage information from the social sensing data, which is generalizable to different types of urban infrastructure damages. Furthermore, our system dynamically leverages the imperfect crowd intelligence to carefully identify and fix the failure cases of AI and provide accurate and timely UIM service.

Crowd-AI Hybrid Systems

Our work is also related to crowd-AI hybrid systems that leverage human intelligence to solve the complex AI-driven computational problems (Goldberg, Wang, and Grant 2017; Zhang et al. 2021; Jarrett et al. 2014; Zhang et al. 2020b; Sener and Savarese 2018; Zhang et al. 2019). For example, Jarrett *et. al.* designed an image complexity quantification mechanism that combines the human and machine intelligence to adaptively optimize the overall performance of the deep learning model for mobile face recognition (Jarrett et al. 2014). Sener *et. al.* proposed a deep core-set selection approach that collects crowd labels from a subset of representative images to retrain the AI models to improve the overall accuracy in natural scene image classification tasks (Sener and Savarese 2018). Zhang *et. al.* designed a crowd-AI hybrid system that leverages crowd intelligence to retrain the AI models and combine crowd labels with AI outputs to troubleshoot and tune the performance of AI algorithms in disaster damage assessment applications (Zhang et al. 2019). Goldberg *et al.* proposed a crowd-assisted text segmentation framework that integrates crowdsourced text labels with a conditional random fields model to accurately segment textual documents (Goldberg, Wang, and Grant 2017). However, those approaches cannot be directly adopted to solve our problem because they often require a sufficient amount of reliable crowd responses to derive accurate crowd labels to optimize the crowd-AI system performance. However, such a large amount of crowd labels are not available in our streaming UIM application due to the "cold start" problem. More importantly, those approaches often use the collected crowd labels to retrain the AI models to capture the dynamics of the streaming data, which could lead to a non-trivial delay to the streaming UIM applications. To the best of our knowledge, the StreamCollab is the first dynamic crowd-AI system to address the streaming UIM problem under a principal streaming analytical framework.

Problem Definition

In this section, we formally present the problem of streaming crowd-AI collaborative urban infrastructure monitoring. We first define a few key terms that will be used in the problem formulation.

Definition 1 Urban Infrastructure Image Stream ($\boldsymbol{X}$): We define $\boldsymbol{X} = \{X_1, X_2, ..., X_T\}$ to be the set of streaming urban infrastructure images collected from social media platforms (e.g., Twitter, Facebook), where each image contains a view of urban infrastructure facilities (as shown in Figure 1). In particular, $X_t \in \boldsymbol{X}$ denotes the infrastructure image collected from the t^{th} timestep and T is the total number of timesteps in a studied streaming UIM application.

Definition 2 Infrastructure Damage-related Features ($\boldsymbol{V}$): We define the infrastructure damage-related features V_t to be the visual features (e.g., building and road cracks, broken traffic signs) in X_t that directly indicate the urban infrastructure damage.

Definition 3 Urban Infrastructure Damage Label ($\boldsymbol{Y}$): An urban infrastructure site is considered as "damaged" (labeled as $Y_t = 1$) if the infrastructure captured in the social media image X_t contains physical damage that can cause potential safety hazards or reduce the functionality of the infrastructure (e.g., Figure 1). Otherwise, the urban infrastructure site captured in the image is considered as "not damaged" (labeled as $Y_t = 0$).

Definition 4 Infrastructure Damage Label Estimated by AI ($\widehat{\boldsymbol{Y^{AI}}}$): We define $\widehat{\boldsymbol{Y^{AI}}}$ to be the infrastructure damage label estimated by AI module of our crowd-AI system. In particular, $\widehat{Y_t^{AI}}$ indicates the estimated infrastructure damage label for X_t at timestep t.

Definition 5 Uncertainty of AI Estimation (U): We first consider the estimation error between the actual and estimated infrastructure damage label $|\overline{Y_t^{AI}} - Y_t|$. $\overline{Y_t^{AI}}$ represents the deep feature generated by a deep neural network for UIM task after the final softmax activation function, which directly indicates the infrastructure damage label in the image X_t (Martins and Astudillo 2016). We observe that such an error often follows a Gaussian distribution (Kendall, Gal, and Cipolla 2018):

$$|\overline{Y_t^{AI}} - Y_t| \sim \mathcal{N}(0, U_t^2) \tag{1}$$

where U_t indicates the estimation uncertainty that represents the standard deviation of the estimation error $|\overline{Y_t^{AI}} - Y_t|$. In addition, we define $\boldsymbol{U} = \{U_1, U_2, ..., U_T\}$ to be the estimation uncertainty for images in $\boldsymbol{X}$.

Definition 6 Crowd Query (Q): We define a crowd query Q as a crowdsourcing task where a subset of images in social media data stream $\boldsymbol{X}$ are dynamically sent to the crowdsourcing platforms to ask the crowd workers to label the potential infrastructure damages. In particular, our crowd-AI system asks a set of J crowd workers to mark the infrastructure damage label for each image in the crowd query.

Definition 7 Crowd Query Ratio (θ): We define θ to be an application-specific parameter that indicates the amount of urban infrastructure-related social media images that are dynamically added to the crowd query based on the performance and budget trade-off in a streaming UIM application.

Definition 8 Infrastructure Damage Label Marked by Crowd ($\widehat{\boldsymbol{Y^{CI}}}$): We define $\widehat{\boldsymbol{Y^{CI}}}$ to be the infrastructure damage label marked by crowd worker from the crowdsourcing platforms. In particular, $\widehat{Y_t^{CI_j}}$ indicates the infrastructure damage label marked by the j^{th} crowd worker CI_j in crowd query Q for X_t.

Definition 9 Infrastructure Damage Label Identified by Crowd-AI system ($\widehat{\boldsymbol{Y}}$): We define $\widehat{\boldsymbol{Y}}$ to be the infrastructure damage label identified by our crowd-AI collaboration system by dynamically fusing the inputs from both AI and crowd (i.e., $\widehat{\boldsymbol{Y^{AI}}}$ and $\widehat{\boldsymbol{Y^{CI}}}$). We will discuss the detailed design on how to derive the accurate $\widehat{\boldsymbol{Y}}$ in the next section. In particular, $\widehat{Y_t}$ indicates the identified infrastructure damage label for X_t at timestep t.

The goal of our problem is to dynamically fuse the AI and crowd intelligence to accurately identify the urban infrastructure damage in the streaming social media images on-the-fly. Using the above definitions, our problem is formally defined as follows:

$$\arg\max \Pr(\widehat{Y_t} = Y_t \mid X_t, Q, \theta), \forall\, 1 \le t \le T \tag{2}$$

where $\widehat{Y_t}$ and Y_t are the estimated and ground-truth label for an urban infrastructure image X_t at timestep t. This problem is challenging because it is difficult to dynamically fuse the uncertain AI and imperfect crowd intelligence to accurately detect infrastructure damage from the UIM data streams. In this paper, we develop a *StreamCollab* framework to address those challenges, which is elaborated in the next section.

Solution

The overview of the StreamCollab is shown in Figure 2. In particular, it consists of two main modules: 1) *Uncertainty-driven Dynamic Quality Estimation (UDQE)* and 2) *Streaming Crowd Knowledge Fusion (SCKF)*. First, the *UDQE* module develops a dynamic deep uncertainty-aware estimation network that estimates the infrastructure damage labels of incoming social media images and infers the uncertainty of the estimation results to detect the failure cases of AI. Then, the *SCKF* module designs a streaming crowd knowledge fusion engine that leverages the imperfect crowd intelligence to effectively fix the failure cases of AI on-the-fly through a novel recursive estimation model.

Uncertainty-driven Dynamic Quality Estimation (UDQE)

In this subsection, we present the uncertainty-driven dynamic quality estimation network design in StreamCollab to dynamically estimate the urban infrastructure damage labels of input images and quantify the uncertainty of the estimation results. An overall design of our UDQE module is shown in Figure 3. Our UDQE design contains three

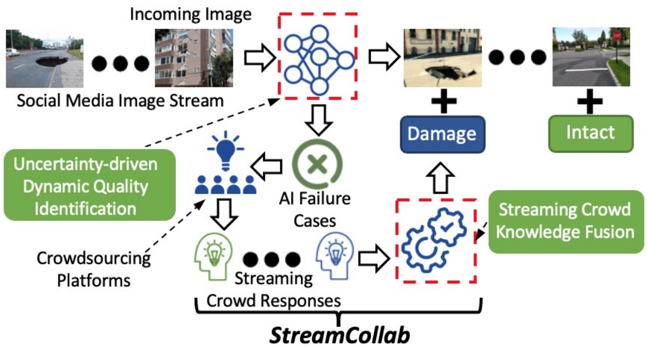


Figure 2: Overview of StreamCollab Framework

core subnetwork modules: a feature extraction subnetwork (*FES*), a damage estimation subnetwork (*DES*), and an uncertainty inference subnetwork (*UIS*). The *FES* first extracts the infrastructure damage related visual features $\boldsymbol{V}$ from the input images $\boldsymbol{X}$. The extracted visual features are forwarded simultaneously to *DES* and *UIS*. In particular, *DES* leverages the extracted visual features $\boldsymbol{V}$ to identify the infrastructure damage labels $\widehat{\boldsymbol{Y^{AI}}}$ for $\boldsymbol{X}$. Meanwhile, *UIS* works in parallel with *DES* to quantify the uncertainty $\boldsymbol{U}$ of the estimated results $\widehat{\boldsymbol{Y^{AI}}}$ generated by *DES*. To the best of our knowledge, the *UDQE* is the first AI-based approach that designs a duo-branch uncertainty estimation network to dynamically detect the failure cases of AI in streaming UIM applications. The optimal instances of our *UDQE* network will be used to estimate the infrastructure damage labels of the streaming social media data and infer the uncertainty of the estimation results. In this subsection, we first present the detailed network architecture for the *UDQE* network and then discuss the dedicated network optimization process to obtain the optimal instances of all subnetworks in *UDQE*. In particular, we first formally define the *FES*, *UIS*, and *DES* as follows:

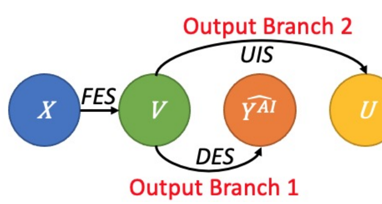


Figure 3: Overview of Uncertainty-driven Dynamic Quality Estimation Design

Definition 10 Feature Extraction Subnetwork (*FES*): We define *FES* as a feature extraction subnetwork to extract the infrastructure damage related features from the input images as follows:

$$\boldsymbol{V} = \mathit{FES}(\boldsymbol{X}) \tag{3}$$

We show the layer-wise architecture of the *FES* in Figure 4(A). In particular, the *FES* contains a ImageNet pretrained convolutional neural network (e.g., VGG) to provide sufficient network depth for complex visual feature extraction.

Definition 11 Damage Estimation Subnetwork (*DES*): We define *DES* as a damage estimation subnetwork that leverages the visual features $\boldsymbol{V}$ extracted by *FES* to identify if there is any infrastructure damage reported in $\boldsymbol{X}$ as follows:

$$\widehat{\boldsymbol{Y^{AI}}} = DES(\boldsymbol{V}) \tag{4}$$

We show the layer-wise architecture of the *DES* in Figure 4 (B). The *DES* includes a stack of dense layers for infrastructure damage estimation.

Definition 12 Uncertainty Inference Subnetwork (*UIS*): We define *UIS* as an uncertainty inference subnetwork to infer the uncertainty of the infrastructure damage estimation results $\widehat{\boldsymbol{Y^{AI}}}$ using the the visual features $\boldsymbol{V}$ extracted by *FES* as follows:

$$\boldsymbol{U} = UIS(\boldsymbol{V}) \tag{5}$$

We show the layer-wise architecture of the *UIS* in Figure 4 (C). The *UIS* includes a stack of dense layers to infer the uncertainty $\boldsymbol{U}$ of the estimation results $\widehat{\boldsymbol{Y^{AI}}}$.

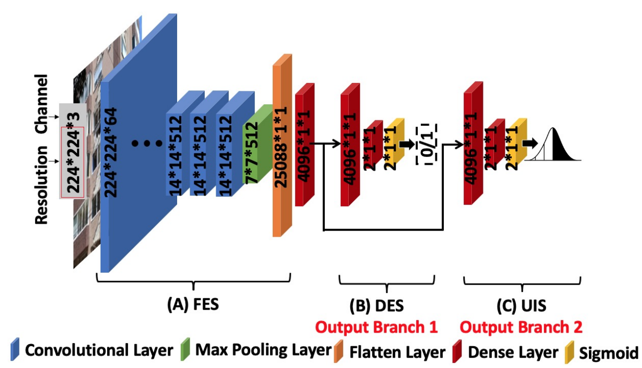


Figure 4: Examples of Layer-wise UDQE Architecture

Given the three core subnetwork modules in our *UDQE* design, the next question is how to obtain the optimal instances of all core subnetworks that maximize the infrastructure damage estimation accuracy while correctly quantifying the uncertainty of the estimation results. In our *UDQE* design, we introduce two loss functions to address the above question. We first define the estimation loss function for *FES* and *DES* as:

$$\mathcal{L}^{\text{ES}}_{FES,DES}: \min\left(\boldsymbol{Y}\log\widehat{\boldsymbol{Y^{AI}}} + (\boldsymbol{1}-\boldsymbol{Y})\log(\boldsymbol{1}-\widehat{\boldsymbol{Y^{AI}}})\right) \tag{6}$$

where $\boldsymbol{Y}$ and $\widehat{\boldsymbol{Y^{AI}}}$ indicate the actual and estimated infrastructure damage label, respectively. In particular, we use the cross-entropy loss function in $\mathcal{L}^{\text{ES}}_{FES,DES}$ to calculate the difference between the actual and estimated infrastructure damage labels. The loss function ensures the *FES* and *DES* work collaboratively to identify the infrastructure damage related visual features and accurately estimate the infrastructure damage labels of the input images.

Our next loss function design focuses on how to learn the accurate uncertainty estimation of the infrastructure damage labels generated by *DES*. Recall that the difference between the actual and estimated infrastructure damage label (i.e., $|\overline{\boldsymbol{Y^{AI}}} - \boldsymbol{Y}|$) follows the Gaussian Distribution (i.e., $\mathcal{N}(\boldsymbol{0}, \boldsymbol{U}^2)$) (Kendall, Gal, and Cipolla 2018). We can obtain the log-likelihood function for $|\overline{\boldsymbol{Y^{AI}}} - \boldsymbol{Y}|$ as:

$$\begin{aligned} &\log \mathcal{L}(\boldsymbol{0}, \boldsymbol{U}; |\overline{\boldsymbol{Y^{AI}}} - \boldsymbol{Y}|) \\ &= -\frac{1}{2}(log||\boldsymbol{U}||_2^2 + \frac{1}{||\boldsymbol{U}||_2^2}||\overline{\boldsymbol{Y^{AI}}} - \boldsymbol{Y}||_2^2 + log2\pi) \end{aligned} \tag{7}$$

We can further transform the log-likelihood function to the uncertainty loss function $\mathcal{L}^{\text{UN}}_{FES,UIS}$ for the *FES* and *UIS* through the function negation as:

$$\begin{aligned} &\mathcal{L}^{\text{UN}}_{FES,UIS}: \\ &\min \left(\frac{1}{2}(log||\boldsymbol{U}||_2^2 + \frac{1}{||\boldsymbol{U}||_2^2}||\overline{\boldsymbol{Y^{AI}}} - \boldsymbol{Y}||_2^2 + log2\pi) \right) \end{aligned} \tag{8}$$

Therefore, we can maximize the log-likelihood function $\log \mathcal{L}(\boldsymbol{0}, \boldsymbol{U}; |\overline{\boldsymbol{Y^{AI}}} - \boldsymbol{Y}|)$ to obtain the accurate uncertainty $\boldsymbol{U}$ by minimizing the $\mathcal{L}^{\text{UN}}_{FES,UIS}$.

Finally, we combine the two loss functions to obtain the overall loss function $\mathcal{L}^{Overall}_{FES,DES,UIS}$ that ensures the *UDQE* module generates the accurate infrastructure damage estimation results $\widehat{\boldsymbol{Y^{AI}}}$ and the uncertainty estimation $\boldsymbol{U}$ as:

$$\mathcal{L}^{Overall}_{FES,DES,UIS} : \mathcal{L}^{\text{ES}}_{FES,DES} + \mathcal{L}^{\text{UN}}_{FES,UIS} \tag{9}$$

We adopt the ADAM optimizer (Kingma and Ba 2014) to optimize $\mathcal{L}^{Overall}_{FES,DES,UIS}$ and obtain the optimal instances (FES^*, DES^*, UIS^*) in the *UDQE* module. The optimal instances are then used to obtain the infrastructure damage results $\widehat{Y_t^{AI}}$ and uncertainty estimation U_t for each incoming image X_t from the social media image stream as:

$$(\widehat{Y_t^{AI}}, U_t) = (DES^*(FES^*(X_t)), UIS^*(FES^*(X_t))) \tag{10}$$

We observe that a higher value in uncertainty estimation $\boldsymbol{U}$ indicates that the AI model is less certain about the estimation result, which indicates the learned infrastructure damage label is more likely to be inaccurate (Emami-Naeini, Akhter, and Rock 1988). We determine whether to add an incoming image X_t to the crowd query based on its uncertainty estimation U_t through active data selection (Abdar et al. 2021). If the image X_t is not selected for the crowd query Q, we use the label $\widehat{Y_t^{AI}}$ estimated by our *UDQE* module as the final output $\widehat{Y_t}$ of our StreamCollab framework for image X_t at timestep t.

Streaming Crowd knowledge Fusion (SCKF)

In the previous subsection, we present the *UDQE* module that identifies the failure cases of AI in the streaming UIM application. Our next question is how to dynamically obtain accurate and timely human intelligence from unreliable crowd workers to fix the failure cases of AI. Unlike urban infrastructure labels obtained from domain experts in infrastructure management, the crowd annotations are often noisy and inconsistent due to the intrinsic uncertainty of crowd workers (e.g., lack of professional knowledge in infrastructure or civil engineering, conflicting responses from different crowd workers) (Hansson and Ludwig 2019). To that end, we develop a streaming crowd knowledge fusion engine that recursively derives accurate infrastructure damage labels from the imperfect crowd responses to fix the failure cases of AI. In particular, we first define a key term that will be used in our *SCKF* module:

Definition 13 Crowd Intelligence Fusion Window (*CW*):We define the *CW* to be a sliding window that includes the most recent I images added to the crowd query Q. In particular, we define $CW = \{X_1, X_2, ..., X_I\}$, where X_i represents the i^{th} image in *CW* and I is the size of *CW*.

Similar to the online video applications that often use a local data buffer to ensure the smooth streaming video service, the *CW* in our model is designed to buffer a set of images with infrastructure damage labels marked by the crowd workers for dynamic crowd knowledge fusion.

Given the above definition, we formulate a maximum likelihood estimation (MLE) problem to derive the *unknown* infrastructure damage labels $\boldsymbol{Y}$ for images in CW by leveraging the imperfect crowd responses from crowd workers with *unknown* reliability as:

$$\Pr\left((\widehat{\boldsymbol{Y^{CI_1}}}, \widehat{\boldsymbol{Y^{CI_2}}}, ..., \widehat{\boldsymbol{Y^{CI_J}}}) | \boldsymbol{Y} \right) \tag{11}$$

where $\widehat{\boldsymbol{Y^{CI_j}}}$ indicates the infrastructure damage labels marked by the j^{th} crowd worker in for images in the crowd query Q. We further define the likelihood function $\mathbb{L}(\Omega; \Delta, Z)$ of our MLE problem as follows:

$$\begin{aligned} &\mathbb{L}(\Omega; \Delta, Z) = \mathbb{L}(\Omega; (\widehat{\boldsymbol{Y^{CI_1}}}, \widehat{\boldsymbol{Y^{CI_2}}}, ..., \widehat{\boldsymbol{Y^{CI_J}}}), \boldsymbol{Y}) \\ &= \prod_{i=1}^{I} \Bigg(\prod_{j=1}^{J} \alpha_j^{+\phi_{i,j}^{+}} \times \alpha_j^{-\phi_{i,j}^{-}} \\ &\times (1 - \alpha_j^{+} - \alpha_j^{-})^{(1 - \phi_{i,j}^{+} - \phi_{i,j}^{-})} \times d \times z_i \\ &+ \prod_{j=1}^{J} \beta_j^{+\phi_{i,j}^{-}} \times \beta_j^{-\phi_{i,j}^{+}} \\ &\times (1 - \beta_j^{+} - \beta_j^{-})^{(1 - \phi_{i,j}^{+} - \phi_{i,j}^{-})} \times (1 - d) \times (1 - z_i) \Bigg) \end{aligned} \tag{12}$$

The above likelihood function represents the likelihood of the observed data Ω (i.e., infrastructure damage labels marked by different crowd workers in current sliding window) and the value of hidden variables Z (i.e., the actual infrastructure damage label for each studied image) given the estimated parameter Ω. We further summarize the detailed explanations of the parameters in the $\mathbb{L}(\Omega; \Delta, Z)$ in Table 1.

Given the MLE problem formulated above, the next key question is how can we solve the MLE problem in the streaming UIM application to fuse the imperfect crowd responses to address the failure cases of AI. To address this question, we derive a recursive expectation maximization (EM) solution to solve the formulated MLE problem. In estimation theory (Wang et al. 2013), the estimation parameter

Notations	Definitions/Explanations
I	size of the crowd intelligence fusion window
J	number of crowd workers
$\phi^+_{i,j}$ & $\phi^-_{i,j}$	indicator variables that are set to be 1 when a crowd worker C_j marks the infrastructure damage label in image X_i to be 1 and 0, respectively
α^+_j & α^-_j	conditional probability that a crowd worker CI_j marks the infrastructure damage label to be 1 and 0 given the actual infrastructure damage label is 1, respectively
β^+_j & β^-_j	conditional probability that a crowd worker CI_j marks the infrastructure damage label to be 1 and 0 given the actual infrastructure damage label is 0, respectively
d	prior probability that the infrastructure damage label of a randomly image is 1
z_i	probability that the infrastructure damage label of image X_i is 1
Ω	estimation parameter of the MLE model, where $\Omega = \{\alpha^+_j, \alpha^-_j, \beta^+_j, \beta^-_j; d\}\, for\, j = 1, 2..., J$
Δ	observed variable of the model, where $\Delta = (\widehat{\boldsymbol{Y^{CI_1}}}, \widehat{\boldsymbol{Y^{CI_2}}}, ..., \widehat{\boldsymbol{Y^{CI_J}}})$
Z	hidden variable of the MLE model, which indicates the infrastructure damage label $\boldsymbol{Y}$ for all studied image

Table 1: Notations in Streaming Crowd Knowledge Fusion

of an MLE problem can be recursively updated in consecutive timesteps by considering the streaming data input as follows:

$$\Omega_{t+1} = \Omega_t + \frac{I_c(\Omega_t)^{-1}\Gamma(X_{t+1}, \Omega_t)}{t+1} \tag{13}$$

where Ω_{t+1} indicate the estimation parameters Ω_t and Ω_{t+1} at two consecutive timestep t and $t+1$, respectively. X_{t+1} indicates the image that is newly added to the crowd intelligence fusion window *CW* at timestep $t+1$. The estimation parameter Ω_{t+1} are used to calculate the updated estimation of the infrastructure damage label for each image in *CW*. $I_c(\Omega_t)^{-1}$ represents the inverse of Fisher information of the estimation parameter Ω_t at timestep t. $\Gamma(X_{t+1}, \Omega_t)$ indicates the score vector of the observed data X_{t+1} given estimation parameter Ω_t at timestep t. The above streaming solution recursively updates the estimation parameter Ω to dynamically derive the accurate infrastructure damage labels and fix the failure cases of AI. To solve the problem, we derive the inverse of Fisher information $I_c(\Omega_t)^{-1}$ and the score vector $\Gamma(X_{t+1}, \Omega_t)$ and plug them into Equation (13) to recursively derive the estimation parameters Ω (i.e., α^+_j,α^-_j, β^+_j,β^-_j) using the above equation as follows:

$$\begin{aligned}
&\alpha_j^{+\,t+1} = \alpha_j^{+\,t} + \frac{1}{(1-\alpha_j^{-\,t}) \times I \times d \times (t+1)} \times \\
&\left(\sum_{i \in G^{t+1}_{j,+}} z_i^{t+1} \times (1 - \alpha_j^{+\,t} - \alpha_j^{-\,t}) - \sum_{i \in G^{t+1}_{j,\emptyset}} z_i^{t+1} \times \alpha_j^{+\,t} \right) \\
&\alpha_j^{-\,t+1} = \alpha_j^{-\,t} + \frac{1}{(1-\alpha_j^{+\,t}) \times I \times d \times (t+1)} \times \\
&\left(\sum_{i \in G^{t+1}_{j,-}} z_i^{t+1} \times (1 - \alpha_j^{+\,t} - \alpha_j^{-\,t}) - \sum_{i \in G^{t+1}_{j,\emptyset}} z_i^{t+1} \times \alpha_j^{-\,t} \right) \\
&\beta_j^{+\,t+1} = \beta_j^{+\,t} + \frac{1}{(1-\beta_j^{-\,t}) \times I \times (1-d) \times (t+1)} \times \\
&\left(\sum_{i \in G^{t+1}_{j,-}} (1 - z_i^{t+1}) \times (1 - \beta_j^{+\,t} - \beta_j^{-\,t}) - \sum_{i \in G^{t+1}_{j,\emptyset}} (1 - z_i^{t+1}) \times \beta_j^{+\,t} \right) \\
&\beta_j^{-\,t+1} = \beta_j^{-\,t} + \frac{1}{(1-\beta_j^{+\,t}) \times I \times (1-d) \times (t+1)} \times \\
&\left(\sum_{i \in G^{t+1}_{j,+}} (1 - z_i^{t+1}) \times (1 - \beta_j^{+\,t} - \beta_j^{-\,t}) - \sum_{i \in G^{t+1}_{j,\emptyset}} (1 - z_i^{t+1}) \times \beta_j^{-\,t} \right)
\end{aligned} \tag{14}$$

where $G^{t+1}_{j,+}$ and $G^{t+1}_{j,-}$ indicates the set of images where the infrastructure damage labels are marked as 1 and 0 by the crowd worker C_j, respectively. $G^{t+1}_{j,\emptyset}$ indicates the set of images that are not marked by the crowd worker C_j. In addition, we observe that z_i^{t+1} is unknown and can be approximated as follows:

$$z_i^{t+1} \approx \frac{P^{t+1}_{i,+} \times d}{P^{t+1}_{i,+} \times d + P^{t+1}_{i,-} \times (1-d)} \tag{15}$$

where $P^{t+1}_{i,+}$ and $P^{t+1}_{i,-}$ can be computed as follows:

$$\begin{aligned}
P^{t+1}_{i,+} &= \prod_{j=1}^{J} \Big(\frac{R^{t+1}_{j,+}}{R^{t}_{j,+}} \times \alpha_j^{+\,t}\Big)^{\phi^+_{i,j}} \times \Big(\frac{R^{t+1}_{j,-}}{R^{t}_{j,-}} \times \alpha_j^{-\,t}\Big)^{\phi^-_{i,j}} \\
&\times \Big(1 - \big(\frac{R^{t+1}_{j,+}}{R^{t}_{j,+}} \times \alpha_j^{+\,t}\big)^{\phi^+_{i,j}} - \big(\frac{R^{t+1}_{j,-}}{R^{t}_{j,-}} \times \alpha_j^{-\,t}\big)^{\phi^-_{i,j}}\Big)^{(1-\phi^+_{i,j}-\phi^-_{i,j})} \\
P^{t+1}_{i,-} &= \prod_{j=1}^{J} \Big(\frac{R^{t+1}_{j,-}}{R^{t}_{j,-}} \times \beta_j^{+\,t}\Big)^{\phi^-_{i,j}} \times \Big(\frac{R^{t+1}_{j,+}}{R^{t}_{j,+}} \times \beta_j^{-\,t}\Big)^{\phi^+_{i,j}} \\
&\times \Big(1 - \big(\frac{R^{t+1}_{j,-}}{R^{t}_{j,-}} \times \beta_j^{+\,t}\big)^{\phi^+_{i,j}} - \big(\frac{R^{t+1}_{j,+}}{R^{t}_{j,+}} \times \beta_j^{-\,t}\big)^{\phi^-_{i,j}}\Big)^{(1-\phi^+_{i,j}-\phi^-_{i,j})}
\end{aligned} \tag{16}$$

where $R^{t/t+1}_{j,+}$ and $R^{t/t+1}_{j,-}$ indicate the number of damage severity labels marked by a crowd worker CI_j in CW as 1 and 0 at time step $t/t+1$, respectively.

Finally, we can dynamically infer the infrastructure damage label for each image in the crowd intelligence fusion window from the dynamically updated z_i. In particular, we define the inferred infrastructure damage label by our recursive EM solution as:

Definition 14 Inferred Infrastructure Damage Label by CI ($\overline{\boldsymbol{Y^{CI}}}$): We define $\overline{Y_i^{CI}}$ to be the estimated infrastructure damage label for image X_i in *CW*. In particular, we set the $\overline{Y_i^{CI}}$ as 1 when $z_i > 0.5$ and 0 otherwise.

Finally, we use the inferred infrastructure damage label $\overline{Y_i^{CI}}$ to replace the label generated by the *UDQE* module for the image X_i in CW to fix the failure cases of AI.

Summary of StreamCollab Framework

Finally, we summarize the StreamCollab framework in Algorithm 1. In particular, StreamCollab includes two main phases in performing the streaming UIM task by exploring the collaborative AI and crowd intelligence as follows:

Model pre-training phase: The objective of this phase is to pre-train an optimized uncertainty-driven dynamic quality estimation network instance (FES^*, DES^* and UIS^*) that will be used to dynamically estimate the infrastructure damage labels and detect the failure cases of AI in the next phase.

Streaming UIM phase: Given the learned optimized FES^*, DES^* and UIS^*, the objective of this phase is to identify the failure cases of AI by selecting the images with high uncertainty and adding those images to the crowd query Q. For the images that are not added to Q, we take the infrastructure damage label estimated by our AI module $\widehat{\boldsymbol{Y^{AI}}}$ as the output $\widehat{\boldsymbol{Y}}$ of our StreamCollab framework. For the images in the crowd query Q, our SCKF module recursively derives the accurate crowd intelligence $\overline{\boldsymbol{Y^{CI}}}$ to fix the failure cases of AI, which is used as the output $\widehat{\boldsymbol{Y}}$ of our StreamCollab framework.

Algorithm 1 StreamCollab Framework Summary

▷ **Model Pre-training Phase**
initialize FES (Definition 10)
initialize DES (Definition 11)
initialize UIS (Definition 12)
for each epoch **do**
  for each batch **do**
    optimize FES, DES and UIS (Equation (9))
  end for
end for
obtain FES^*, DES^*, and UIS^*
▷ **Steaming UIM Phase**
for each incoming X_t (timestep t) **do**
  obtain $\widehat{Y_t^{AI}}$ and U_t using FES^*, DES^*, and UIS^* (Equation (10))
  if add X_t to Q **then**
    obtain $\widehat{Y_t^{CI}}$ from crowdsourcing platform
    add X_t to CW
    calculate z_i using Equation (15)
    calculate α_j^+,α_j^-, β_j^+,β_j^- using (14)
    derive $\overline{Y_t^{CI}}$ using Definition 14
    set $\overline{Y_t^{CI}}$ as $\widehat{Y}_t$
    output $\widehat{Y}_t$
  else
    set $\widehat{Y_t^{AI}}$ as $\widehat{Y}_t$
    output $\widehat{Y}_t$
  end if
end for

Evaluation

In this section, we evaluate the performance of the StreamCollab system using a real-world streaming UIM dataset collected from online social media. The results show that StreamCollab consistently outperforms both state-of-the-art AI-only and crowd-AI baselines in correctly identifying the damaged urban infrastructure under various types of evaluation scenarios.

Dataset and Crowdsourcing Platform

Urban Infrastructure Images Dataset on Social Media We first describe the real-world social media dataset of urban infrastructure images used in our study. In particular, we collect 1,200 urban infrastructure related images from Twitter using GetOldTweets[2]. The ground truth labels for urban infrastructure damage are annotated by domain experts for the evaluation purpose. In particular, it consists of 612 (51.0%) and 588 (49.0%) images of urban infrastructure with damages and without damages, respectively. In addition, we keep the ratio of training to testing data as 6:4. The training dataset is used to train all compared AI models for urban infrastructure damage identification.

Crowdsourcing Platform We use Amazon Mechanical Turk (AMT) to obtain crowd intelligence. AMT is one of the largest crowdsourcing platform that can provides 24/7 crowdsourcing services with a large amount of crowd workers worldwide. In the crowdsourcing task, we recruit the crowd workers with an overall task approval rate > 95% and have finished at least 1000 approved tasks to ensure the crowd label quality. We pay $\$0.05$ to each worker per image in our experiment. We follow the IRB protocol approved for this project. In our experiment, we study a diversified set of crowd query settings, where we vary the crowd query ratio from 10% to 20% and vary the number of crowd workers from 2 to 5.

Baseline and Settings

We compare StreamCollab with a set of representative deep learning based and crowd-AI hybrid solutions for urban infrastructure damage detection.

- **MobileNet** (Howard et al. 2017): MobileNet is a light weight convolutional neural network architecture that can efficiently learn visual representations from social media image content for infrastructure detection in UIM applications.
- **DenseNet** (Huang et al. 2017): DenseNet is a dense convolutional network framework that connects all the layers in the neural network to efficiently propagate latent visual features for identifying damages in the urban infrastructure images.
- **VGG** (Li et al. 2018): VGG is a deep convolutional neural network based solution that learns the latent visual features in the urban infrastructure images for damage detection.

[2]https://github.com/Mottl/GetOldTweets3

Category	Algorithm	θ = 10%			θ = 15%			θ = 20%		
		$\mathcal{K}$-Score	MCC	F1-Score	$\mathcal{K}$-Score	MCC	F1-Score	$\mathcal{K}$-Score	MCC	F1-Score
Random	Random	0.0174	0.0174	0.5082	0.0114	0.0114	0.5059	0.0168	0.0168	0.5091
AI-Only	MobileNet	0.4974	0.5056	0.7463	0.5290	0.5290	0.7645	0.5242	0.5251	0.7621
	DenseNet	0.5098	0.5170	0.7528	0.5298	0.5318	0.7643	0.5379	0.5389	0.7686
	VGG	0.5424	0.5451	0.7704	0.5440	0.5514	0.7710	0.5660	0.5667	0.7830
Crowd-AI	Hybrid Para	0.5117	0.5128	0.7566	0.5334	0.5350	0.7669	0.5465	0.5486	0.7728
	Deep Active	0.4693	0.4723	0.7342	0.5321	0.5348	0.7658	0.5443	0.5495	0.7715
	CrowdLearn	0.5428	0.5465	0.7705	0.5491	0.5568	0.7732	0.5657	0.5673	0.7832
Our Model	**StreamCollab**	**0.6076**	**0.6100**	**0.8037**	**0.6245**	**0.6275**	**0.8120**	**0.6368**	**0.6394**	**0.8184**

Table 2: Performance Comparisons on UIM Classification Accuracy

- **Hybrid Para** (Jarrett et al. 2014): Hybrid Para is a crowdsourcing approach that combines the human and machine intelligence to adaptively optimize the overall performance of the damage detection model of the UIM.
- **Deep Active** (Sener and Savarese 2018): Deep Active is a deep active learning based crowd-AI scheme that adopts a core-set selection approach to select a subset of representative images that share the deep visual features with all studied images from crowd workers to annotate and improve the performance of damage identification in UIM.
- **CrowdLearn** (Zhang et al. 2019): CrowdLearn is a crowd-AI hybrid system that leverages crowd intelligence to retrain the AI models and combines crowd labels with AI outputs to troubleshoot and tune the performance of AI algorithms.

In our experiments, we keep the same inputs to all compared schemes for a fair comparison. In particular, the inputs to a scheme include: 1) the studied social media images; 2) the ground-truth labels of images in the training dataset, and 3) the labeled images from the crowd workers. In particular, we retrain the AI only baselines using the crowd labels for a fair comparison. We also include a random baseline for UIM tasks that randomly decides if there are any infrastructure damages in a studied image. Note that we do not include the crowd-only baselines that task the crowd workers to examine the infrastructure damage of all studied images due to the infeasible labeling costs in the real world applications (Pan et al. 2016). In our experiment, our StreamCollab model is implemented using PyTorch 1.1.0 libraries [3] and is trained on the NVIDIA Quadro RTX 6000 GPUs. We also optimize all hyper-parameters through the Adam optimizer (Kingma and Ba 2014) using a learning rate of 10^{-5}. In addition, we set the batch size to be 20 and train the model over 100 epochs.

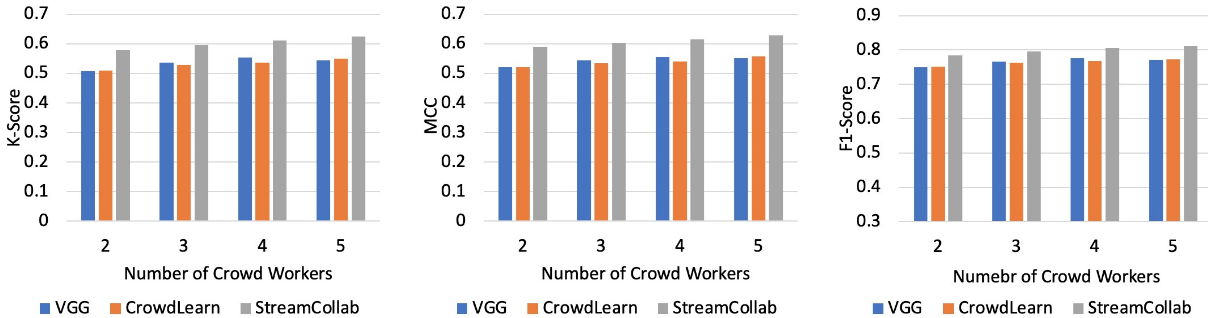
To evaluate the performance of all compared schemes, we use three metrics that are widely adopted to evaluate the performance of image classification tasks in image processing and machine learning (Chicco and Jurman 2020): 1) *Cohen's kappa Score ($\mathcal{K}$-Score)* (Artstein and Poesio 2008), 2) *Matthews Correlation Coefficient (MCC)* (Jurman, Riccadonna, and Furlanello 2012), and 3) *F1-score*. The higher values of the three metrics indicate better UIM performance.

[3]https://pytorch.org

Evaluation Results

UIM Classification Accuracy Comparison In the first set of experiments, we compare the accuracy of all compared schemes in identifying the infrastructure damage of input social media images. In particular, we study the performance of all compared schemes by varying the crowd query ratio θ from 10% to 20%, which achieve a reasonable balance between the number of crowd responses and the query cost. In addition, we set the number of crowd workers to be 5. The evaluation results are presented in Table 2. We observe that our StreamCollab scheme consistently outperforms all compared baselines when the crowd query ratio changes. For example, the performance gain of StreamCollab compared to the best-performing baseline (i.e., CrowdLearn) when the crowd query ratio $\theta = 10\%$ on $\mathcal{K}$-Score, MCC, and F-Score are 6.48%, 6.35%, and 3.32%, respectively. Such performance gains mainly come from the fact that our SreamCollab develops a dynamic deep uncertainty-aware estimation network to effectively detect the failure cases of AI and designs a streaming crowd knowledge fusion model that derives accurate infrastructure damage labels from imperfect crowd responses to fix the detected failure cases of AI. We also observe that the performance of our StreamCollab improves when we increase the crowd query ratio. This is because, with a larger crowd query ratio, our StreamCollab can effectively leverage more crowd responses to fix problematic AI cases to improve the overall performance of our StreamCollab framework.

Computational Efficiency Comparison In the second set of experiments, we compare the computational cost of all compared schemes (except the trivial *random* baseline) in the studied streaming UIM application. We define the computational cost as the average computational time required to identify the infrastructure damage label of an input image. To ensure a fair compassion, we evaluate all schemes



(a) $\mathcal{K}$-Score (b) MCC (c) F1-Score

Figure 5: Robustness of StreamCollab Scheme

using the same NVIDIA Quadro RTX 6000 GPU. The results are shown in Table 3. We observe that our StreamCollab scheme takes significantly less time to accomplish the streaming UIM task than the compared baselines under different evaluation settings. This is because the compared baselines needs to retrain their models to capture the dynamics of the streaming data by leveraging the labels from crowd workers. In contrast, our StreamCollab designs a streaming crowd-AI collaborating solution that identifies and fixes the AI failure cases on-the-fly without requiring any additional model retraining.

Robustness of StreamCollab Scheme In the third set of experiments, we evaluate the robustness of StreamCollab scheme over the different number of crowd workers. In our experiment, we vary the number of crowd worker from 2 to 5. In addition, we set the crowd query ratio to be 15%. We also compare the performance of our StreamCollab with the best-performing AI-only baseline (i.e., VGG) and the best-performing crowd-AI baseline (i.e., CrowdLearn). The evaluation results are presented in Figure 5. We observe that the performance of our StreamCollab is relatively stable as the crowd worker number changes. We also observe that our StreamCollab consistently outperforms the best-performing baselines over all three evaluation metrics when the number of crowd worker changes. The above results demonstrate the robustness and effectiveness of our StreamCollab scheme in effectively leveraging the imperfect crowd intelligence from different number of crowd workers to fix the failure cases of AI on-the-fly through a novel recursive estimation model.

Algorithm	$\theta = 10\%$	$\theta = 15\%$	$\theta = 20\%$
MobileNet	0.1250	0.1778	0.2246
DenseNet	0.1510	0.2149	0.2708
VGG	0.1624	0.2379	0.2983
Hybrid Para	3.1531	3.1533	3.1536
DeepActive	0.1207	0.1723	0.2214
CrowdLearn	0.1614	0.2323	0.2910
StreamCollab	**0.0225**	**0.0227**	**0.0232**

Table 3: Computational Time Comparisons (Seconds)

Conclusion

In this paper, we present a StreamCollab framework to solve a streaming UIM problem in social sensing applications. StreamCollab addresses two key challenges, namely, *streaming crowd-AI collaboration* and *uncertainty in streaming crowd intelligence*. In particular, we propose a streaming crowd-AI collaborative system that dynamically explores the collaborative intelligence from AI and crowd to detect infrastructure damages reported in streaming social sensing data. The StreamCollab framework achieves clear performance gains in terms of both UIM accuracy and computational efficiency compared to state-of-the-art AI-only and crowd-AI baselines in a real-word streaming UIM application. We believe that StreamCollab provides useful insights to design new crowd-AI collaboration systems for real-world streaming social sensing and smart city applications (e.g., intelligent transportation, disaster response, and misinformation detection).

Acknowledgments

This research is supported in part by the National Science Foundation under Grant No. IIS-2008228, CNS-1845639, CNS-1831669, Army Research Office under Grant W911NF-17-1-0409. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

Abdar, M.; Pourpanah, F.; Hussain, S.; Rezazadegan, D.; Liu, L.; Ghavamzadeh, M.; Fieguth, P.; Cao, X.; Khosravi, A.; Acharya, U. R.; et al. 2021. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion* .

Alavi, A. H.; and Buttlar, W. G. 2019. An overview of smartphone technology for citizen-centered, real-time and scalable civil infrastructure monitoring. *Future Generation Computer Systems* 93: 651–672.

Artstein, R.; and Poesio, M. 2008. Inter-coder agreement for computational linguistics. *Computational Linguistics* 34(4): 555–596.

Capponi, A.; Fiandrino, C.; Kantarci, B.; Foschini, L.; Kliazovich, D.; and Bouvry, P. 2019. A survey on mobile crowdsensing systems: Challenges, solutions, and opportunities. *IEEE communications surveys & tutorials* 21(3): 2419–2465.

Cervone, G.; Schnebele, E.; Waters, N.; Moccaldi, M.; and Sicignano, R. 2017. Using social media and satellite data for damage assessment in urban areas during emergencies. In *Seeing cities through big data*, 443–457. Springer.

Chicco, D.; and Jurman, G. 2020. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC genomics* 21(1): 6.

Dutta, J.; Gazi, F.; Roy, S.; and Chowdhury, C. 2016. AirSense: Opportunistic crowd-sensing based air quality monitoring system for smart city. In *2016 IEEE SENSORS*, 1–3. IEEE.

Emami-Naeini, A.; Akhter, M. M.; and Rock, S. M. 1988. Effect of model uncertainty on failure detection: the threshold selector. *IEEE Transactions on Automatic Control* 33(12): 1106–1115.

Figueiras, P.; Herga, Z.; Guerreiro, G.; Rosa, A.; Costa, R.; and Jardim-Gonçalves, R. 2018. Real-time monitoring of road traffic using data stream mining. In *2018 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, 1–8. IEEE.

Goldberg, S.; Wang, D. Z.; and Grant, C. 2017. A probabilistically integrated system for crowd-assisted text labeling and extraction. *Journal of Data and Information Quality (JDIQ)* 8(2): 1–23.

Hand, M. 2017. Visuality in social media: Researching images, circulations and practices. *The SAGE handbook of social media research methods* 217–231.

Hansson, K.; and Ludwig, T. 2019. Crowd dynamics: Conflicts, contradictions, and community in crowdsourcing. *Computer Supported Cooperative Work (CSCW)* 28(5): 791–794.

Harris, D. K.; Alipour, M.; Acton, S. T.; Messeri, L. R.; Vaccari, A.; and Barnes, L. E. 2017. The citizen engineer: Urban infrastructure monitoring via crowd-sourced data analytics. In *Structures Congress 2017*, 495–510.

Howard, A. G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; and Adam, H. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* .

Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely Connected Convolutional Networks. In *CVPR*, volume 1, 3.

Huang, J.; Duan, N.; Ji, P.; Ma, C.; Ding, Y.; Yu, Y.; Zhou, Q.; Sun, W.; et al. 2018. A crowdsource-based sensing system for monitoring fine-grained air quality in urban environments. *IEEE Internet of Things Journal* 6(2): 3240–3247.

Jarrett, J.; Saleh, I.; Blake, M. B.; Malcolm, R.; Thorpe, S.; and Grandison, T. 2014. Combining human and machine computing elements for analysis via crowdsourcing. In *10th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing*, 312–321. IEEE.

Jurman, G.; Riccadonna, S.; and Furlanello, C. 2012. A comparison of MCC and CEN error measures in multi-class prediction. *PloS one* 7(8): e41882.

Kankanamge, N.; Yigitcanlar, T.; Goonetilleke, A.; and Kamruzzaman, M. 2019. Can volunteer crowdsourcing reduce disaster risk? A systematic review of the literature. *International journal of disaster risk reduction* 35: 101097.

Kendall, A.; Gal, Y.; and Cipolla, R. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7482–7491.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Laubis, K.; Konstantinov, M.; Simko, V.; Gröschel, A.; and Weinhardt, C. 2019. Enabling crowdsensing-based road condition monitoring service by intermediary. *Electronic Markets* 29(1): 125–140.

Li, X.; Caragea, D.; Zhang, H.; and Imran, M. 2018. Localizing and quantifying damage in social media images. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 194–201. IEEE.

Li, Y.; Gao, J.; Meng, C.; Li, Q.; Su, L.; Zhao, B.; Fan, W.; and Han, J. 2016. A survey on truth discovery. *ACM Sigkdd Explorations Newsletter* 17(2): 1–16.

Lin, Y.; and Li, R. 2020. Real-time traffic accidents post-impact prediction: Based on crowdsourcing data. *Accident Analysis & Prevention* 145: 105696.

Mandel, T.; Best, J.; Tanaka, R. H.; Temple, H.; Haili, C.; Carter, S. J.; Schlechtinger, K.; and Szeto, R. 2020. Using the Crowd to Prevent Harmful AI Behavior. *Proceedings of the ACM on Human-Computer Interaction* 4(CSCW2): 1–25.

Martins, A.; and Astudillo, R. 2016. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *International Conference on Machine Learning*, 1614–1623.

Mavridis, P.; Gross-Amblard, D.; and Miklós, Z. 2016. Using hierarchical skills for optimized task assignment in knowledge-intensive crowdsourcing. In *Proceedings of the 25th International Conference on World Wide Web*, 843–853.

Mei, Q.; and Gül, M. 2019. A crowdsourcing-based methodology using smartphones for bridge health monitoring. *Structural Health Monitoring* 18(5-6): 1602–1619.

Mejova, Y.; Weber, I.; and Fernandez-Luque, L. 2018. Online health monitoring using Facebook advertisement audience estimates in the United States: evaluation study. *JMIR public health and surveillance* 4(1): e30.

Oshri, B.; Hu, A.; Adelson, P.; Chen, X.; Dupas, P.; Weinstein, J.; Burke, M.; Lobell, D.; and Ermon, S. 2018. Infrastructure quality assessment in africa using satellite imagery and deep learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 616–625.

Pan, Z.; Yu, H.; Miao, C.; and Leung, C. 2016. Efficient collaborative crowdsourcing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Sener, O.; and Savarese, S. 2018. Active Learning for Convolutional Neural Networks: A Core-Set Approach. In *International Conference on Learning Representations*.

Seto, T.; and Sekimoto, Y. 2019. Trends in citizen-generated and collaborative urban infrastructure feedback data: Toward citizen-oriented infrastructure management in Japan. *ISPRS International Journal of Geo-Information* 8(3): 115.

Wang, D.; Abdelzaher, T.; and Kaplan, L. 2015. *Social sensing: building reliable systems on unreliable data*. Morgan Kaufmann.

Wang, D.; Abdelzaher, T.; Kaplan, L.; and Aggarwal, C. C. 2013. Recursive fact-finding: A streaming approach to truth estimation in crowdsourcing applications. In *2013 IEEE 33rd international conference on distributed computing systems*, 530–539. IEEE.

Wang, D.; Amin, M. T.; Li, S.; Abdelzaher, T.; Kaplan, L.; Gu, S.; Pan, C.; Liu, H.; Aggarwal, C. C.; Ganti, R.; et al. 2014. Using humans as sensors: an estimation-theoretic perspective. In *IPSN-14 proceedings of the 13th international symposium on information processing in sensor networks*, 35–46. IEEE.

Wang, D.; Kaplan, L.; and Abdelzaher, T. F. 2014. Maximum likelihood analysis of conflicting observations in social sensing. *ACM Transactions on Sensor Networks (ToSN)* 10(2): 1–27.

Wang, D.; Kaplan, L.; Le, H.; and Abdelzaher, T. 2012. On truth discovery in social sensing: A maximum likelihood estimation approach. In *Information Processing in Sensor Networks (IPSN), 2012 ACM/IEEE 11th International Conference on*, 233–244. IEEE.

Wang, D.; Szymanski, B. K.; Abdelzaher, T.; Ji, H.; and Kaplan, L. 2019a. The age of social sensing. *Computer* 52(1): 36–45.

Wang, L.; Yang, C.; Yu, Z.; Liu, Y.; Wang, Z.; and Guo, B. 2019b. CrackSense: A CrowdSourcing Based Urban Road Crack Detection System. In *2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, 944–951. IEEE.

Wang, Y.; Jia, X.; Jin, Q.; and Ma, J. 2017. Mobile crowdsourcing: framework, challenges, and solutions. *Concurrency and Computation: Practice and experience* 29(3): e3789.

Zhang, D.; Ma, Y.; Hu, X. S.; and Wang, D. 2020a. Toward Privacy-Aware Task Allocation in Social Sensing-Based Edge Computing Systems. *IEEE Internet of Things Journal* 7(12): 11384–11400.

Zhang, D.; Zhang, Y.; Li, Q.; Plummer, T.; and Wang, D. 2019. Crowdlearn: A crowd-ai hybrid system for deep learning-based damage assessment applications. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, 1221–1232. IEEE.

Zhang, D. Y.; Han, R.; Wang, D.; and Huang, C. 2016. On robust truth discovery in sparse social media sensing. In *Big Data (Big Data), 2016 IEEE International Conference on*, 1076–1081. IEEE.

Zhang, D. Y.; Huang, Y.; Zhang, Y.; and Wang, D. 2020b. Crowd-assisted disaster scene assessment with human-ai interactive attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 2717–2724.

Zhang, J.; and Wang, D. 2015. Duplicate report detection in urban crowdsensing applications for smart city. In *2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)*, 101–107. IEEE.

Zhang, Y.; Zong, R.; Kou, Z.; Shang, L.; and Wang, D. 2021. CollabLearn: An Uncertainty-Aware Crowd-AI Collaboration System for Cultural Heritage Damage Assessment. *IEEE Transactions on Computational Social Systems* 1–15. doi:10.1109/TCSS.2021.3109143.

Zhao, X.; Wang, N.; Han, R.; Xie, B.; Yu, Y.; Li, M.; and Ou, J. 2018. Urban infrastructure safety system based on mobile crowdsensing. *International journal of disaster risk reduction* 27: 427–438.