Synergizing Point-Based CCTV and Wide-Area Remote Sensing Intelligence for Adaptive Flood Monitoring in Bandung

Bayulodie Vallianto[1,2*], Masahiko Nagai[1], Yusuf Cahyadi[3]

[1] Graduate School of Sciences and Technology for Innovation, Yamaguchi University, Ube, Japan

[2] National Research and Innovation Agency (BRIN), Jakarta, Indonesia

[3] Bandung Command Center, Bandung, Indonesia

*bayulodie.val@gmail.com

Abstract *Urban flood monitoring remains a challenge for cities where rapid inundation during the rainy season disrupts transportation and risks public safety. This study addresses the critical gap in urban flood monitoring with the aid of combining real-time point-based closed-circuit television (CCTV) streams with wide-area remote sensing intelligence for Bandung City, Indonesia. Our framework continuously processes feeds from 10 flood-prone geotagged CCTV locations, extracting frames every 20 seconds through OpenCV and classifying water severity (dry, wet, or flood) using a pre-trained MobileNetV2 and fine-tuned on 3,058 actual frames that achieve promising performance with 94% classification accuracy in controlled tests. When floods are detected, the pipeline triggers elevation-guided spatial interpolation using a Digital Elevation Model (DEM), modeling flood spread along low-lying roads through Inverse Distance Weighting. This method estimates water surface elevation across a 500-meter radius around each flood point. The interpolated output traces probable inundation extent by comparing water elevations against ground-level DEM values, providing emergency responders with actionable flood spread forecasts. This integration strategically bridges the temporal gain of CCTV (real-time point data) and the spatial intelligence of remote sensing (DEM terrain analysis), overcoming their individual shortcomings. As an ongoing studies initiative, the framework is being refined for operational assessment by means of the Bandung Command Center, with future work focusing on field deployment.*

Keywords: Flood severity classification, Geospatial fusion, MobileNetV2, Real-time interpolation, Bandung

Introduction

Urban flooding represents a significant and escalating risk to cities globally, particularly in the context of rapid urbanization and the increasing prevalence of extreme weather events. Effective hazard assessment and management strategies are crucial to mitigate the impacts of such disasters. Cities like Bandung, Indonesia, exemplify this vulnerability, facing persistent flood challenges that underscore the urgent need for robust flood management policies and innovative monitoring solutions. The Bandung Flood of 2016, for instance, highlighted the critical role of timely information sharing during such events. Addressing these challenges demands advanced, near-real-time monitoring capabilities to reduce damages and enhance urban resilience.

Traditional approaches to urban flood monitoring often rely on disparate methods, each presenting inherent limitations. Ground camera-based computer vision techniques, including Closed-Circuit Television (CCTV) networks, offer valuable insights for urban flood monitoring and cyber surveillance for flood disasters, providing high-frequency temporal data from fixed, point-based perspectives (Dhaya & Kanthavel, 2022, p. 1; Jun et al., 2024, p. 7; Lo et al., 2015, pp. 5–6). However, these systems inherently lack comprehensive spatial intelligence beyond their immediate field of view, making it difficult to understand the dynamic spatial change in runoff or the full extent of inundation (Lo et al., 2015, p. 5). Conversely, remote sensing techniques are widely recognized for their application in flood management, particularly for identifying flood-prone areas and providing wide-area synoptic views (Lo et al., 2015, p. 4; Zhang et al., 2023, p. 3). Yet, these methods often face challenges such as low temporal resolution, cloud obstruction during storms, and delays in data processing, which can hinder their utility for rapid, real-time event response and continuous monitoring at fixed points in urban areas (Lo et al., 2015, p. 4).

A critical gap therefore exists in effectively integrating the immediacy and high-frequency temporal data from point-based visual sensing systems, like CCTV, with the extensive spatial intelligence required for dynamic, wide-area flood extent forecasting. Current methods struggle to combine real-time, localized observations with the broader spatial context necessary for adaptive and comprehensive flood monitoring, particularly in rapidly evolving urban flood scenarios.

This study directly addresses this critical gap by introducing an integrated framework that strategically merges real-time point-based CCTV video analysis with wide-area geospatial intelligence for adaptive urban flood monitoring in Bandung City. We leverage existing public CCTV infrastructure from flood-prone, geotagged locations. Utilizing computer vision techniques, video frames are extracted at regular intervals and processed by a fine-tuned deep learning model, such as MobileNetV2 (Zhou et al., 2021, p. 3) al., 2018】, to classify water severity in near-real-time. Upon detecting a "flood" event, our framework initiates an elevation-driven spatial interpolation process. By employing a high-resolution Digital Elevation Model (DEM) (Zhou et al., 2021, p. 3), we utilize Inverse Distance Weighting (IDW) (Shepard, 1968) to estimate the water surface height around the detected point. Comparing this interpolated water surface with the ground elevation from the DEM allows us to model the probable inundation extent within a defined radius and trace flood spread along topographic features. This core innovation bridges the temporal advantage of CCTV with the spatial intelligence derived from DEM analysis.

Our integrated framework provides actionable flood spread forecasts within minutes of detection, offering crucial situational awareness for emergency responders and decision-makers in Bandung. By synergizing ubiquitous real-time CCTV streams with fundamental geospatial terrain data, we overcome the individual deficiencies of each method, delivering a cost-effective, scalable, and adaptive solution for cities with existing camera networks. This work directly contributes to enhancing urban resilience and disaster risk reduction by providing a rapid and effective analysis tool for flood hazard mapping and management measures (Zhang et al., 2023, pp. 24–25; Zhou et al., 2021, p. 12).

Literature Review

Urban flooding has emerged as a pressing global concern, with its prevalence increasing rapidly due to extreme weather events and the complexities of urban environments . Effective flood management policies and robust monitoring systems are essential for mitigating the severe impacts of these disasters, which can cripple infrastructure, paralyze transportation, and endanger public safety . Cities in developing nations, particularly those experiencing rapid urbanization and inadequate drainage management, are acutely vulnerable. Bandung City, Indonesia, serves as a prime example, facing annual floods that necessitate comprehensive disaster response and risk mitigation strategies (Setiadi et al., 2023, p. 2). The challenges in Bandung include land conversion, poor waste and drainage management, critical watersheds, and illegal buildings, all contributing to the city's susceptibility to flooding (Setiadi et al., 2023, p. 2,8,11). The need for innovative, near-real-time monitoring solutions is paramount to provide decision-makers with timely and accurate information for effective flood control and emergency response (Lo et al., 2015, p. 6; Zhang et al., 2023, pp.24–25; Zhou et al., 2021, p. 12).

Ground Camera-Based Flood Monitoring

Ground camera-based computer vision (CV) techniques, including Closed-Circuit Television (CCTV) networks, have gained significant attention for urban flood monitoring due to their ability to provide real-time, high-frequency temporal data (Dhaya & Kanthavel, 2022, p. 1; Jun et al., 2024, p. 7; Lo et al., 2015, pp. 5–6). These systems are capable of capturing dynamic field information, which is crucial for understanding rapidly evolving flood events (Lo et al., 2015, p. 5). Research objectives in this domain often include flood depth estimation, flood detection, identification of flooded areas, and surface water velocity measurement (Jun et al., 2024, p. 7). Various CV techniques are employed, such as image

classification, object detection, segmentation, edge detection, and tracking (Jun et al., 2024, pp. 7–8).

For instance, studies have explored using Convolutional Neural Networks (CNN) for classifying water levels from surveillance videos (Dhaya & Kanthavel, 2022, p. 1). The process typically involves collecting video frames, extracting features (e.g., using Histogram of Oriented Gradients (HoG)), enhancing frames to remove noise, and then classifying water levels using CNNs (Dhaya & Kanthavel, 2022, p. 1). Deep learning techniques, including Mask R-CNN, have been applied for object detection and segmentation in flood management contexts (Jun et al., 2024, p. 7,18). Other research focuses on estimating spatial-temporal distribution of urban street ponding levels from surveillance videos based on computer vision (Jun et al., 2024, p. 18). Visual sensing systems can automatically analyze real-time field images to determine flood severity using image processing with virtual markers, providing critical information to decision-makers for rapid warnings and responses (Lo et al., 2015, p. 6).

Despite these advancements, a primary limitation of ground camera-based systems is their inherent point-based nature. While they offer high temporal resolution at specific locations, they lack comprehensive spatial coverage and intelligence beyond their fixed field of view (Lo et al., 2015, p. 5). This makes it challenging to assess the overall flood extent or predict flood propagation across a wider urban area, which is vital for holistic disaster management.

Remote Sensing and Geospatial Intelligence for Flood Monitoring

Remote sensing technologies offer a complementary approach by providing wide-area spatial intelligence for flood monitoring and management. These techniques are valuable for obtaining synoptic perspectives and identifying flood-prone areas, their distributions, extents, and flow paths (Lo et al., 2015, p. 4; Zhang et al., 2023, p. 3; Zhou et al., 2021, p. 3,12). Digital Elevation Models (DEMs) are widely used for flood prediction and monitoring, enabling hydrological analyses such as depression extraction (DE) and Topographic Wetness Index (TWI) assessments to characterize inundated areas and surface flow directions (Zhou et al., 2021, p. 3,12). Satellite remote sensing, particularly using Synthetic Aperture Radar (SAR), can acquire data regardless of weather conditions, facilitating the detection of flooded zones in urban areas (Lo et al., 2015, p. 4).

The integration of various data sources, often referred to as multi-source data fusion, is increasingly recognized for urban waterlogging risk identification (Zhang et al., 2023, pp. 1-2,25). This involves processing urban meteorological, geographic, and municipal

engineering information in a unified manner to form a deep fusion of multi-data layers, combined with hydrodynamic theory to simulate surface runoff and predict waterlogging extent and depth (Zhang et al., 2023, pp. 1-2,25). Such methods can provide pre-disaster reference information and statistics about affected areas, supporting rapid and effective delineation of risk control areas and timely release of early warning information (Zhang et al., 2023, p. 3,24-25). Inverse Distance Weighting (IDW) is a well-established spatial interpolation technique used to estimate values at unmeasured locations based on surrounding measured points, making it suitable for estimating water surface heights from discrete observations (Shepard, 1968).

However, remote sensing methods also have limitations. They often suffer from low temporal resolution due to infrequent satellite revisits, cloud obstruction during storms, and delays in data processing, which can render them unsuitable for rapid event response (Lo et al., 2015, p. 4). Furthermore, continuous monitoring at fixed points in small urban rivers is difficult with remote sensing technology, and the accuracy of these methods can be dependent on the raster accuracy of the DEM and digital maps (Lo et al., 2015, p. 4; Zhou et al., 2021, p. 12).

The Need for Integrated Solutions

The review of existing literature highlights a clear dichotomy: ground camera-based systems excel in providing real-time, high-frequency *temporal* data at specific points, while remote sensing and geospatial intelligence offer comprehensive *spatial* coverage over wide areas. Both approaches, individually, present significant limitations for dynamic, real-time urban flood extent forecasting. Ground cameras lack the spatial context, and remote sensing often lacks the temporal immediacy required for rapid event response in fast-changing urban flood scenarios (Lo et al., 2015, pp. 4–5).

Therefore, a critical research gap lies in the effective integration of these two powerful, yet individually limited, monitoring paradigms. The challenge is to synergize the high-frequency temporal observations from ubiquitous point-based visual sensing systems with the extensive spatial intelligence derived from remote sensing and high-resolution topographic data. Such an integrated framework would overcome the deficiencies of each method, enabling a more adaptive, accurate, and timely understanding of urban flood dynamics, which is essential for enhancing urban resilience and disaster risk reduction efforts.

Methodology

This research employs data acquisition and processing, flood severity classification, and a comprehensive urban flood analysis structure that includes a three-step process for flood mapping.

1. Data Acquisition and Pre-processing

The initial stage of this method included systematic acquisition and pre-processing of video data from strategically located surveillance infrastructure.

a. Video Data Source

There are over 300 Closed-Circuit Television (CCTV) cameras that were already deployed within Bandung City, but for this research only 10 geotagged CCTV cameras in flood-prone areas were used. These cameras constantly stream video feeds, providing a dynamic view record of environmental conditions. The geotagged nature of these cameras is important to connect visual observations with geographical coordinates, which is necessary for successive mapping processes.

b. Frame Extraction and Processing

Video feeds from CCTV cameras were processed systematically at intervals of 20 seconds. This interval was chosen to provide a critical balance between the need for high temporal resolution to detect rapid changes and the computational efficiency required for a near-real-time system. Also, effectively capture transient hydrological events, especially during the period of extreme rainfall when the development of flood conditions is the most likely. Video frame processing, including extraction and initial analysis, was done using the OpenCV library (Bradski, 2000), a widely recognized open-source computer vision and machine learning software library. The primary purpose of the move was to extract relevant frames that later reflect different water levels and conditions for the severity classification of floods.

2. Flood Severity Classification Model Development

To accurately identify and classify flood events from the extracted video frames, a deep learning-based classification model was developed and fine-tuned.

a. Model Architecture

The core of the flood severity classification system is a fine-tuned MobileNetV2 architecture (Sandler et al., 2018). MobileNetV2 was chosen due to its lightweight, efficiency and effectiveness in mobile and embedded vision applications, which is potentially suitable for real-time processing requirements. The pre-trained weights of

MobileNetV2 was used as a starting point, and the model was then adapted to the specific function of flood severity classification.

b. Data Preparation and Imbalance Mitigation

The core of the fine-tuning process utilized a dataset of 3,058 real frames carefully extracted from CCTV video feeds. These frames were manually annotated and distributed into three distinct classes: 1,093 for 'dry' (Level 0), 1,558 for 'wet' (Level 1), and 407 for 'flood' (Level 2). Given the significant class imbalance, particularly the under-representation of the critical 'flood' class, we employed a weighted categorical cross-entropy loss function to prevent the MobileNetV2 model from becoming biased toward the majority classes. This function assigns a greater penalty to misclassifications of the minority class, forcing the model to learn more effectively from the limited 'flood' examples. The determined weights for each class were as follows: Level 0 (Dry) at 0.932, Level 1 (Wet) at 0.654, and Level 2 (Flood) at 2.504.

c. Two-Staged Fine-Tuning Strategy

The MobileNetV2 model was fine-tuned using a two-staged training strategy to ensure stable convergence and optimal transfer learning performance:

Table 1: Two-Staged Fine-Tuning Strategy.

Stage	Duration	Focus	Learning Rate (LR)
Stage 1	20 Epochs	Classification Head Only	Standard (9.0000e-04 – 1.0000e-04)
Stage 2	30 Epochs	Full Network Unfrozen	Very Small (4.5000e-05 – 5.0000e-06)

In Stage 1, only the newly added classification head was trained for 20 epochs while the MobileNetV2 base weights remained frozen, a step crucial for stabilizing the output layer. The learning rate for this stage began at 9.0000e-04 and was gradually decreased by a scheduler, concluding at 1.0000e-04. Subsequently, in Stage 2, the entire network was unfrozen and fine-tuned for an additional 30 epochs. This stage utilized a very small initial learning rate of 4.5000e-05 to prevent catastrophic forgetting, which was then gradually reduced to 5.0000e-06 to preserve the valuable pre-trained features while adapting them precisely to the specific flood classification data.

Training utilized several standard callbacks to regulate the process, including Early Stopping to prevent overfitting, Model Checkpointing to save the best weights, and

ReduceLROnPlateau and a dedicated Learning Rate Scheduler to dynamically control the optimization process.

d. Classification Thresholds

The trained MobileNetV2 model classifies each processed frame into one of the three flood severity categories based on the predetermined visual criteria:

- Dry: This category is assigned when there is no visible water on the ground or surface within the camera viewing area.
- Wet: This classification indicates the presence of surface moisture, such as moisture or puddle, but without any significant accumulation of standing water that will disrupt or cause damage.
- Flood: This category reflects the presence of accumulated standing water, indicating a flood occurs that can potentially cause disruption or damage.

3. Inundation Mapping Pipeline

When detecting a "flood" event by the classification model, an automated flood mapping pipeline is triggered to portray the spatial extent of floods.

a. Digital Elevation Model (DEM) Extraction

Geotagged coordinates of the CCTV camera that detected the flood incident serve as a starting point for spatial analysis. These coordinates are used to extract relevant Digital Elevation Model (DEM) data to understand terrain topography. DEM data was obtained from DEMNAS provided by Indonesia's Geospatial Information Agency (BIG) with a resolution of 8 meters which provides sufficient details for local scale urban flood mapping.

b. Water Surface Elevation (WSE) Interpolation

To estimate the height of the water surface in the affected area, the Inverse Distance Weighting (IDW) interpolation method was employed. The IDW is a determinable interpolation method that estimates unknown values based on the values of the surrounding known points, which gives more weight to points closer to the estimated location. The formula used for WSE interpolation is:

$$WSE(x,y) = \frac{\sum_{i=1}^{n} \frac{z_i}{d_i^2}}{\sum_{i=1}^{n} \frac{1}{d_i^2}}$$

Where:

- $WSE(x,y)$ represents the estimated Water Surface Elevation (WSE) at a given coordinate.

- z_i is the known value at a nearby data point i.
- d_i is the distance between the known data point i and the given location (x,y).
- d_i^2 represents the weight of each data point. The use of a squared distance (p=2) ensures that data points closer to the estimation point have a significantly greater influence than those farther away.
- The numerator $(\sum \frac{z_i}{d_i^2})$ is the sum of the known values multiplied by their respective weights.
- The denominator $(\sum \frac{1}{d_i^2})$ is the sum of all the weights, which acts as a normalization factor.

This interpolation is performed using the information of the water level detected as a reference point from the CCTV camera, as well as using the surrounding elevation data from the DEM.

c. Flood Extent Delineation

Flood mapping pipeline involves delineating the flood extent in the final stage. This is obtained by comparing the interpolated Water Surface Elevation (WSE) with the Digital Elevation Model (DEM) elevation. Any area where the WSE is greater than the DEM elevation is classified as inundated. This comparison is made within a radius of 500 meters from the geotagged CCTV camera that detected the flood. This radius was chosen to provide a practical and actionable local map for emergency responders, while minimizing interpolation errors which would increase with greater distance from the single observation point. The output of this stage is a spatial map illustrating the estimated area affected by the flood.

Results

1. Model Performance

The fine-tuned MobileNetV2 model for flood severity classification underwent rigorous evaluation using a dedicated test set.

a. Classification Accuracy

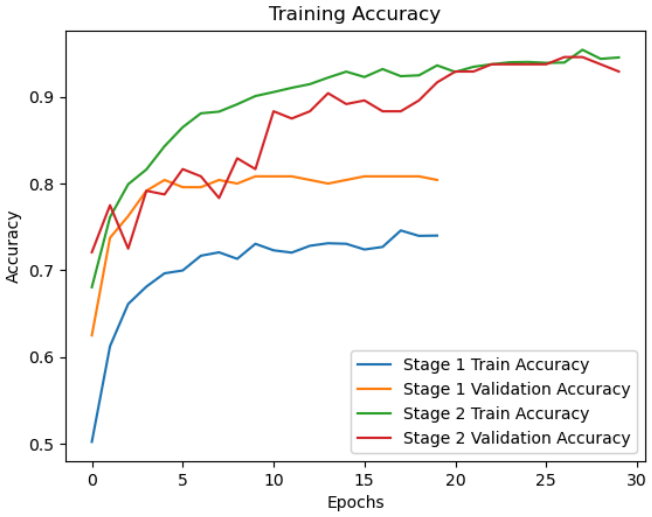

Figure 1: Training Accuracy Progression.

During Stage 1, both training and validation accuracies show a steady rise in the early epochs, with training accuracy reaching approximately 0.74 and validation accuracy stabilizing around 0.80. This indicates that the model quickly captures the general patterns but plateaus due to limited learning capacity or suboptimal initialization.

```
Epoch 1/20
96/96 ━━━━━━━━━━━━━━━━━━━━ 114s 1s/step - accuracy: 0.4482 - loss: 1.2879 - val_accuracy: 0.6250 - val_loss: 0.8086 - learning_rate: 9.0000e-04
Epoch 2/20
96/96 ━━━━━━━━━━━━━━━━━━━━ 95s 986ms/step - accuracy: 0.5832 - loss: 0.9352 - val_accuracy: 0.7375 - val_loss: 0.6258 - learning_rate: 8.1000e-04
Epoch 3/20
96/96 ━━━━━━━━━━━━━━━━━━━━ 95s 991ms/step - accuracy: 0.6529 - loss: 0.8074 - val_accuracy: 0.7625 - val_loss: 0.5656 - learning_rate: 7.2900e-04
Epoch 4/20
96/96 ━━━━━━━━━━━━━━━━━━━━ 95s 986ms/step - accuracy: 0.6892 - loss: 0.7749 - val_accuracy: 0.7917 - val_loss: 0.5305 - learning_rate: 6.5610e-04
Epoch 5/20
96/96 ━━━━━━━━━━━━━━━━━━━━ 95s 983ms/step - accuracy: 0.6907 - loss: 0.7432 - val_accuracy: 0.8042 - val_loss: 0.4896 - learning_rate: 5.9049e-04
Epoch 6/20
96/96 ━━━━━━━━━━━━━━━━━━━━ 94s 975ms/step - accuracy: 0.6892 - loss: 0.6983 - val_accuracy: 0.7958 - val_loss: 0.5107 - learning_rate: 5.3144e-04
Epoch 7/20
96/96 ━━━━━━━━━━━━━━━━━━━━ 94s 979ms/step - accuracy: 0.7084 - loss: 0.6993 - val_accuracy: 0.7958 - val_loss: 0.4994 - learning_rate: 4.7830e-04
Epoch 8/20
96/96 ━━━━━━━━━━━━━━━━━━━━ 94s 975ms/step - accuracy: 0.7233 - loss: 0.6462 - val_accuracy: 0.8042 - val_loss: 0.4997 - learning_rate: 2.1523e-04
Epoch 9/20
96/96 ━━━━━━━━━━━━━━━━━━━━ 94s 976ms/step - accuracy: 0.7037 - loss: 0.6522 - val_accuracy: 0.8000 - val_loss: 0.4925 - learning_rate: 1.9371e-04
Epoch 10/20
96/96 ━━━━━━━━━━━━━━━━━━━━ 95s 983ms/step - accuracy: 0.7268 - loss: 0.6405 - val_accuracy: 0.8083 - val_loss: 0.4858 - learning_rate: 1.7434e-04
Epoch 11/20
96/96 ━━━━━━━━━━━━━━━━━━━━ 95s 982ms/step - accuracy: 0.7187 - loss: 0.6053 - val_accuracy: 0.8083 - val_loss: 0.4804 - learning_rate: 1.5691e-04
Epoch 12/20
96/96 ━━━━━━━━━━━━━━━━━━━━ 96s 994ms/step - accuracy: 0.7201 - loss: 0.6186 - val_accuracy: 0.8083 - val_loss: 0.4797 - learning_rate: 1.4121e-04
Epoch 13/20
96/96 ━━━━━━━━━━━━━━━━━━━━ 95s 984ms/step - accuracy: 0.7508 - loss: 0.5895 - val_accuracy: 0.8042 - val_loss: 0.4802 - learning_rate: 1.2709e-04
Epoch 14/20
96/96 ━━━━━━━━━━━━━━━━━━━━ 95s 984ms/step - accuracy: 0.7337 - loss: 0.6142 - val_accuracy: 0.8000 - val_loss: 0.4814 - learning_rate: 1.1438e-04
Epoch 15/20
96/96 ━━━━━━━━━━━━━━━━━━━━ 94s 978ms/step - accuracy: 0.7177 - loss: 0.6658 - val_accuracy: 0.8042 - val_loss: 0.4803 - learning_rate: 5.1473e-05
Epoch 16/20
96/96 ━━━━━━━━━━━━━━━━━━━━ 94s 973ms/step - accuracy: 0.7212 - loss: 0.6179 - val_accuracy: 0.8083 - val_loss: 0.4702 - learning_rate: 1.0000e-04
Epoch 17/20
96/96 ━━━━━━━━━━━━━━━━━━━━ 94s 974ms/step - accuracy: 0.7318 - loss: 0.6495 - val_accuracy: 0.8083 - val_loss: 0.4674 - learning_rate: 1.0000e-04
Epoch 18/20
96/96 ━━━━━━━━━━━━━━━━━━━━ 94s 971ms/step - accuracy: 0.7311 - loss: 0.6248 - val_accuracy: 0.8083 - val_loss: 0.4678 - learning_rate: 1.0000e-04
Epoch 19/20
96/96 ━━━━━━━━━━━━━━━━━━━━ 94s 972ms/step - accuracy: 0.7463 - loss: 0.6060 - val_accuracy: 0.8083 - val_loss: 0.4656 - learning_rate: 1.0000e-04
Epoch 20/20
96/96 ━━━━━━━━━━━━━━━━━━━━ 93s 970ms/step - accuracy: 0.7424 - loss: 0.5884 - val_accuracy: 0.8042 - val_loss: 0.4629 - learning_rate: 1.0000e-04
```

Figure 2: Stage 1 Training Progression.

In Stage 2, after fine-tuning, the training accuracy continues to improve and surpasses 0.90, while validation accuracy also follows closely, reaching around 0.92. The close alignment between training and validation curves suggests that the model generalizes well without significant overfitting. The accuracy improvements in Stage 2 demonstrate the effectiveness of the staged training strategy.

```
Epoch 1/30
96/96 ━━━━━━━━━━━━━━━━━━━━ 460s 5s/step - accuracy: 0.6459 - loss: 0.8197 - val_accuracy: 0.7208 - val_loss: 0.8138 - learning_rate: 4.5000e-05
Epoch 2/30
96/96 ━━━━━━━━━━━━━━━━━━━━ 431s 4s/step - accuracy: 0.7533 - loss: 0.5564 - val_accuracy: 0.7750 - val_loss: 0.5823 - learning_rate: 4.0500e-05
Epoch 3/30
96/96 ━━━━━━━━━━━━━━━━━━━━ 428s 4s/step - accuracy: 0.7924 - loss: 0.4642 - val_accuracy: 0.7250 - val_loss: 0.8748 - learning_rate: 3.6450e-05
Epoch 4/30
~/Documents/Dev/water_level_classifier/data/dem_bandung.tif : 0.8135 - loss: 0.4287 - val_accuracy: 0.7917 - val_loss: 0.5653 - learning_rate: 3.2805e-05
96/96 ━━━━━━━━━━━━━━━━━━━━ 432s 4s/step - accuracy: 0.8238 - loss: 0.3814 - val_accuracy: 0.7875 - val_loss: 0.6279 - learning_rate: 2.9525e-05
Epoch 6/30
96/96 ━━━━━━━━━━━━━━━━━━━━ 435s 5s/step - accuracy: 0.8574 - loss: 0.3173 - val_accuracy: 0.8167 - val_loss: 0.5384 - learning_rate: 2.6572e-05
Epoch 7/30
96/96 ━━━━━━━━━━━━━━━━━━━━ 423s 4s/step - accuracy: 0.8795 - loss: 0.2911 - val_accuracy: 0.8083 - val_loss: 0.5566 - learning_rate: 2.3915e-05
Epoch 8/30
96/96 ━━━━━━━━━━━━━━━━━━━━ 422s 4s/step - accuracy: 0.8802 - loss: 0.2800 - val_accuracy: 0.7833 - val_loss: 0.6311 - learning_rate: 2.1523e-05
Epoch 9/30
96/96 ━━━━━━━━━━━━━━━━━━━━ 433s 5s/step - accuracy: 0.8932 - loss: 0.2399 - val_accuracy: 0.8292 - val_loss: 0.5234 - learning_rate: 1.9371e-05
Epoch 10/30
96/96 ━━━━━━━━━━━━━━━━━━━━ 430s 4s/step - accuracy: 0.8987 - loss: 0.2302 - val_accuracy: 0.8167 - val_loss: 0.5308 - learning_rate: 1.7434e-05
Epoch 11/30
96/96 ━━━━━━━━━━━━━━━━━━━━ 433s 4s/step - accuracy: 0.9025 - loss: 0.2295 - val_accuracy: 0.8833 - val_loss: 0.3703 - learning_rate: 1.5691e-05
Epoch 12/30
96/96 ━━━━━━━━━━━━━━━━━━━━ 433s 5s/step - accuracy: 0.9108 - loss: 0.2143 - val_accuracy: 0.8750 - val_loss: 0.3676 - learning_rate: 1.4121e-05
Epoch 13/30
96/96 ━━━━━━━━━━━━━━━━━━━━ 430s 4s/step - accuracy: 0.9110 - loss: 0.2100 - val_accuracy: 0.8833 - val_loss: 0.3411 - learning_rate: 1.2709e-05
Epoch 14/30
96/96 ━━━━━━━━━━━━━━━━━━━━ 431s 4s/step - accuracy: 0.9216 - loss: 0.1974 - val_accuracy: 0.9042 - val_loss: 0.3020 - learning_rate: 1.1438e-05
Epoch 15/30
96/96 ━━━━━━━━━━━━━━━━━━━━ 433s 5s/step - accuracy: 0.9364 - loss: 0.1624 - val_accuracy: 0.8917 - val_loss: 0.3201 - learning_rate: 1.0295e-05
Epoch 16/30
96/96 ━━━━━━━━━━━━━━━━━━━━ 433s 5s/step - accuracy: 0.9132 - loss: 0.2170 - val_accuracy: 0.8958 - val_loss: 0.3103 - learning_rate: 9.2651e-06
Epoch 17/30
96/96 ━━━━━━━━━━━━━━━━━━━━ 433s 5s/step - accuracy: 0.9386 - loss: 0.1574 - val_accuracy: 0.8833 - val_loss: 0.2998 - learning_rate: 1.0000e-05
Epoch 18/30
96/96 ━━━━━━━━━━━━━━━━━━━━ 430s 4s/step - accuracy: 0.9226 - loss: 0.1924 - val_accuracy: 0.8833 - val_loss: 0.3065 - learning_rate: 1.0000e-05
Epoch 19/30
96/96 ━━━━━━━━━━━━━━━━━━━━ 429s 4s/step - accuracy: 0.9294 - loss: 0.1764 - val_accuracy: 0.8958 - val_loss: 0.2717 - learning_rate: 1.0000e-05
Epoch 20/30
96/96 ━━━━━━━━━━━━━━━━━━━━ 403s 4s/step - accuracy: 0.9329 - loss: 0.1594 - val_accuracy: 0.9167 - val_loss: 0.2381 - learning_rate: 1.0000e-05
Epoch 21/30
96/96 ━━━━━━━━━━━━━━━━━━━━ 400s 4s/step - accuracy: 0.9309 - loss: 0.1598 - val_accuracy: 0.9292 - val_loss: 0.2154 - learning_rate: 1.0000e-05
Epoch 22/30
96/96 ━━━━━━━━━━━━━━━━━━━━ 400s 4s/step - accuracy: 0.9397 - loss: 0.1528 - val_accuracy: 0.9292 - val_loss: 0.1936 - learning_rate: 1.0000e-05
Epoch 23/30
96/96 ━━━━━━━━━━━━━━━━━━━━ 417s 4s/step - accuracy: 0.9343 - loss: 0.1647 - val_accuracy: 0.9375 - val_loss: 0.1938 - learning_rate: 1.0000e-05
Epoch 24/30
96/96 ━━━━━━━━━━━━━━━━━━━━ 431s 4s/step - accuracy: 0.9444 - loss: 0.1364 - val_accuracy: 0.9375 - val_loss: 0.1839 - learning_rate: 1.0000e-05
Epoch 25/30
96/96 ━━━━━━━━━━━━━━━━━━━━ 425s 4s/step - accuracy: 0.9462 - loss: 0.1445 - val_accuracy: 0.9375 - val_loss: 0.1754 - learning_rate: 1.0000e-05
Epoch 26/30
96/96 ━━━━━━━━━━━━━━━━━━━━ 424s 4s/step - accuracy: 0.9309 - loss: 0.1440 - val_accuracy: 0.9375 - val_loss: 0.1689 - learning_rate: 1.0000e-05
Epoch 27/30
96/96 ━━━━━━━━━━━━━━━━━━━━ 431s 4s/step - accuracy: 0.9314 - loss: 0.1544 - val_accuracy: 0.9458 - val_loss: 0.1638 - learning_rate: 1.0000e-05
Epoch 28/30
96/96 ━━━━━━━━━━━━━━━━━━━━ 430s 4s/step - accuracy: 0.9543 - loss: 0.1058 - val_accuracy: 0.9458 - val_loss: 0.1762 - learning_rate: 1.0000e-05
Epoch 29/30
96/96 ━━━━━━━━━━━━━━━━━━━━ 428s 4s/step - accuracy: 0.9427 - loss: 0.1408 - val_accuracy: 0.9375 - val_loss: 0.1747 - learning_rate: 1.0000e-05
Epoch 30/30
96/96 ━━━━━━━━━━━━━━━━━━━━ 429s 4s/step - accuracy: 0.9448 - loss: 0.1272 - val_accuracy: 0.9292 - val_loss: 0.1673 - learning_rate: 5.0000e-06
```

Figure 3: Stage 2 Training Progression.

The model achieved an overall accuracy of 94% across various test scenarios. This performance was evaluated on a dedicated test dataset comprising 90 frames, which were strictly distinct from the training data. The size of this test set was limited due to the rarity of real flood events in the collected operational footage, yet it represents a diverse set of conditions crucial for validation. This high accuracy indicates a robust ability to correctly classify frames into 'dry', 'wet', or 'flood' categories under diverse conditions.

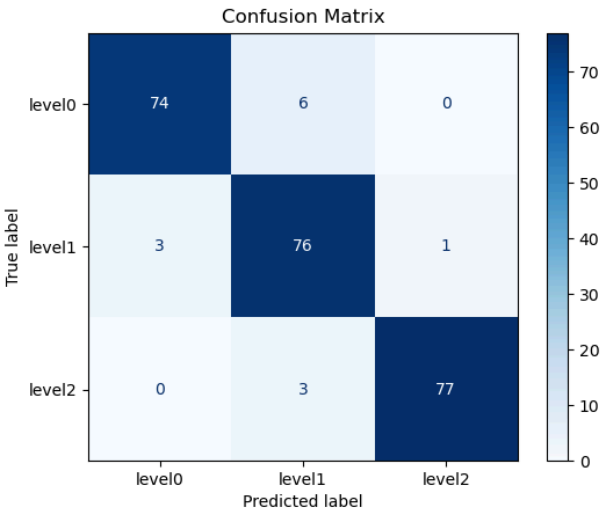

Figure 4: Confusion Matrix.

Furthermore, analysis of the confusion matrix provided deeper insight into the model's classification strengths:

- High Effectiveness: The model was highly effective, correctly classifying a majority of cases for all levels.
- Minimal and Adjacent Errors: Misclassifications were rare and occurred primarily between adjacent levels (Level 0 with Level 1, and Level 1 with Level 2).
- Strong Class Separation: Crucially, the matrix showed a clear strength in distinguishing the extreme classes, with no misclassifications observed between Level 0 (dry) and Level 2 (flood). This strong separation confirms the model's reliability in issuing critical flood alerts.

b. False Positives

Analysis of the model's performance revealed a low rate of false positives, specifically, a false positive rate less than 15% during 'wet-to-flood' transitions. This metric is crucial as it indicates the model's ability to minimize erroneous flood alerts when conditions are merely wet, thereby reducing unnecessary alarms and ensuring the reliability of flood detection.

2. Inundation Mapping Output

The inundation mapping pipeline, activated upon flood detection, produced near-real-time spatial outputs that accurately delineate the flood extent and provide a predictive view of its spread.

a. Near-Real-Time Flood Inundation Mapping

The core of our methodology involves a DEM-guided spatial interpolation technique, which successfully generated flood inundation maps by translating point-based flood detections into estimated flood extents. Specifically, upon detection of a 'flood' event at a geotagged CCTV location, the system provided the critical input needed to manually interpolate the flood spread using the Digital Elevation Model (DEM) data. These maps effectively illustrated the predicted spread of floodwater across the landscape, thus demonstrating the *potential* to provide immediate visual information for emergency response and urban planning. For instance, the system successfully identified the flood point at CCTV location Kopo Citarip, as shown in Figure 5 and generated a DEM-based visualization of the predicted inundation spread, as shown in Figure 6. This visualization confirms the feasibility of the pipeline, with the final automation of the interpolation step remaining an area of continued development.

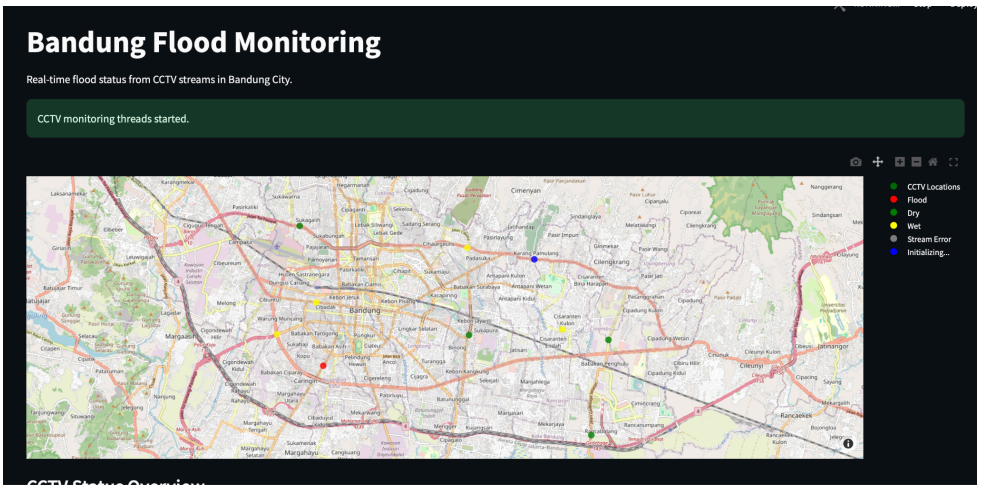

Figure 5: Flood Detection Visualization Map.

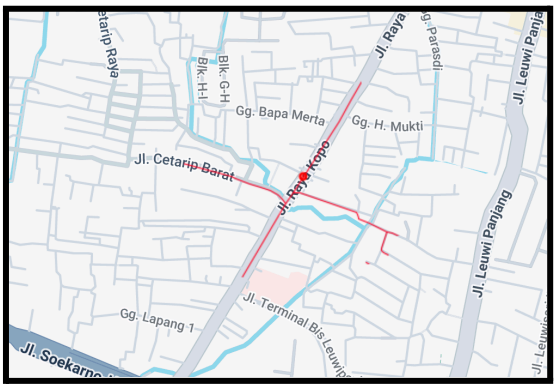

Figure 6: Predicted Road Inundation Spread.

Discussion

This section elaborates on the implications of the framework's performance, highlighting its benefits in terms of system integration and acknowledging the challenges encountered during implementation.

1. **System Integration Benefits**

The integrated system offers several advantages for urban flood management, leveraging the strengths of its individual components.

a. Enhanced Temporal Resolution

The continuous streaming and 20-second interval processing of CCTV feeds enable a remarkable temporal resolution for flood detection. This allows for flood detection cycles as frequent as 3 minutes, providing near real-time updates on evolving flood conditions. Such rapid detection is critical for timely alerts and emergency response coordination in fast-developing flood events.

b. Improved Spatial Intelligence

The integration of geotagged CCTV detections with high-resolution Digital Elevation Models (DEM) significantly enhances spatial intelligence. By contextualizing point-based flood detections within the terrain's topography, the system can generate comprehensive inundation forecasts. This transforms isolated observations into actionable spatial information, illustrating the potential spread and depth of floodwaters.

c. Overcoming Data Imbalance through Weighted Loss Function

A significant challenge in developing robust flood detection models is the scarcity of real captured CCTV frame data for rare flood events. The methodology addresses this by employing a weighted loss function to account for the class imbalance. This function effectively overcomes sample class imbalance, it will compensate for the data imbalance by increasing the weight of the minority class's lost contribution, improving the generalization capabilities and robustness of the classification model.

2. Implementation Challenges

Despite the promising results, the implementation of the system highlighted several challenges that warrant further research and refinement.

a. Terrain Complexity and Hydrodynamic Refinement

The current inundation mapping pipeline, relying on DEM-guided IDW interpolation, provides a static representation of flood extent. However, the complex terrain complexity of urban environments, particularly the intricate network of drainage systems and varying surface permeabilities, necessitates a more dynamic approach. To accurately model water flow and accumulation in such environments, hydrodynamic refinement of the inundation mapping process is required. Incorporating hydrodynamic models would allow for a more accurate simulation of flood propagation, considering factors like flow velocity, friction, and interaction with urban infrastructure.

b. Network stability

During the implementation phase, we encountered several significant challenges that impacted system stability and data retrieval. These issues can be categorized into three primary areas:

- Network and Protocol Errors: We observed frequent network instability, leading to TLS errors such as `connection reset by peer`, `broken pipe`,

and invalidation of the specified session. These issues also manifested as HTTPS errors where the stream terminated prematurely.

- Video Decoding Errors: The primary challenge at the application layer was related to H.264 video streams. We faced a variety of decoding failures, including decoder errors, instances where a left block was unavailable, and more specific issues like `sps_id out of range` and `cabac decode of qscale diff failed`.
- Data Integrity: These combined errors often resulted in incomplete data streams, leading to a final error code indicating the stream had ended prematurely or that the system was unable to fetch a frame.

These challenges highlighted the need for robust error handling and a resilient data retrieval framework.

Conclusion and Future Work

This study successfully developed and demonstrated an integrated system for real-time flood monitoring and inundation mapping in flood-prone urban areas, specifically within Bandung city. By leveraging geotagged CCTV camera feeds, advanced computer vision techniques, and geospatial analysis, the system provides a robust framework for enhancing urban flood resilience.

The flood severity classification model, based on a fine-tuned MobileNetV2 architecture, achieved a commendable 94% accuracy across various test scenarios, while maintaining a low false positive rate of less than 15% during critical wet-to-flood transitions. The implementation of a weighted loss function proved instrumental in overcoming the inherent scarcity and class imbalance of real-world flood event samples. This technique achieved robust generalization by assigning a proportionally greater penalty to misclassifications of the minority 'flood' class, thereby forcing the model to learn more effectively from the limited critical examples.

Upon flood detection, the automated inundation mapping pipeline, utilizing DEM-guided Inverse Distance Weighting (IDW) interpolation, effectively delineated flood extents. This pipeline demonstrated a near-real-time flood detection and efficiently predicted how floodwater will spread across the landscape, validating its ability to translate point-based observations into actionable spatial intelligence. The system's integrated approach offers significant benefits, including high temporal resolution for near-real-time flood detection, improved spatial contextualization via DEM-guided interpolation, and an essential strategy

for mitigating data scarcity through the implementation of a weighted loss function. Overall, this research presents a viable and promising methodology for proactive urban flood management and early warning systems.

Future work will focus on expanding the validation and operational scope of the integrated system. Key research directions include:

- System Validation and Refinement: We plan to validate the flood detection accuracy using Synthetic Aperture Radar (SAR) historical flood maps, which will provide a critical benchmark for the system's spatial output. This will be supported by utilizing external precipitation index data to better contextualize and support the analysis of rainfall intensity.
- Data Enhancement: Efforts will be made to acquire and integrate higher-resolution CCTV data streams to improve the robustness and detail of the computer vision model's classifications.
- Operational Integration: The final phase involves full integration with the Bandung Command Center, transforming the current pipeline into a seamless operational tool. This step will enable a comprehensive flood detection analysis study under real-world emergency conditions.
- Broader Impact: The continued development and deployment of this solution will exemplify how the novel fusion of computer vision and geospatial analytics can effectively bridge critical urban flood monitoring gaps in other Global South cities.

References

Dhaya, R., & Kanthavel, R. (2022). *IASC | Video Surveillance-Based Urban Flood Monitoring System Using a Convolutional Neural Network*. Intelligent Automation & Soft Computing; Tech Science Press. https://doi.org/10.32604/iasc.2022.021538

Jun, S., Jang, H., Kim, S., Lee, J.-S., & Jung, D. (2024). A review of ground camera-based computer vision techniques for flood management. *Computers and Concrete*, *33*(4), 425–443. https://doi.org/10.12989/CAC.2024.33.4.425

Lo, S.-W., Wu, J.-H., Lin, F.-P., & Hsu, C.-H. (2015, August 14). *Visual Sensing for Urban Flood Monitoring*. Sensors; MDPI AG. https://doi.org/10.3390/s150820006

Zhang, Z., Zeng, Y., Huang, Z., Liu, J., & Yang, L. (2023, January 31). *Multi-Source Data Fusion and Hydrodynamics for Urban Waterlogging Risk Identification*. International Journal of Environmental Research and Public Health; MDPI AG. https://doi.org/10.3390/ijerph20032528

Zhou, Q., Su, J., Arnbjerg-Nielsen, K., Ren, Y., Luo, J., Ye, Z., & Feng, J. (2021, May 25). *A*

GIS-Based Hydrological Modeling Approach for Rapid Urban Flood Hazard Assessment. Water; MDPI AG. https://doi.org/10.3390/w13111483

Setiadi, S., Sumaryana, A., Bekti, H., & Sukarno, D. (2023). The flood management policy in Bandung city: Challenges and potential strategies. *Cogent Social Sciences*, *9*(2). https://doi.org/10.1080/23311886.2023.2282434

Bradski, G. (2000). The OpenCV Library. Dr. Dobb's Journal of Software Tools, 120; 122-125.

M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. -C. Chen. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, pp. 4510-4520, doi: 10.1109/CVPR.2018.00474.

Shepard, D. (1968). A two-dimensional interpolation function for irregularly-spaced data. Proceedings of the 1968 ACM National Conference, 517-524.