Improvement of Rainfall Estimation Accuracy Using a Convolutional Neural Network with Convolutional Block Attention Model on Surveillance Camera

Iqbal[1], Adhi Harmoko Saputro[2], Alhadi Bustamam[3], Ardasena Sopaheluwakan[4]
Department of Physics, Faculty of Mathematics and Natural Sciences, Universitas Indonesia, Depok, Indonesia[1, 2]
Department of Mathematics, Faculty of Mathematics and Natural Sciences, Universitas Indonesia, Depok, Indonesia[3]
Meteorological, Climatological, and Geophysical Agency, Jakarta, Indonesia[4]

***Abstract*—Accurate rainfall estimation is essential for various applications, including transportation management, agriculture, and climate modeling. Traditional measurement methods, such as rain gauges and radar systems, often face challenges due to limited spatial resolution and susceptibility to environmental interferences. These constraints affect the ability of the model to deliver high-resolution, real-time rainfall data, allowing the model to be challenging to capture localized variations effectively. Therefore, this study aimed to introduce a hybrid deep learning architecture that combined a Convolutional Neural Network (CNN) with a Convolutional Block Attention Module (CBAM) to improve rainfall intensity estimation using images captured by surveillance cameras. The proposed model was evaluated using standard datasets and previous unseen images collected at different times of the day, including morning, noon, afternoon, and night, to assess its toughness against temporal variations. The experimental results showed that VGG-CBAM architecture performed better than ResNet (Residual Network)-CBAM across all evaluation metrics, achieving a coefficient of determination (R^2) of 0.93 compared to 0.89. Furthermore, when tested on unseen images captured at different periods, the model showed strong generalization capability, with correlation values (R) ranging from 0.77 to 0.98. These results signified the effectiveness of the proposed method in improving the accuracy and adaptability of image-based rainfall estimation, offering a scalable and high-resolution alternative to conventional measurement methods.**

***Keywords*—Rainfall; surveillance camera; hybrid deep learning; CBAM**

I. INTRODUCTION

Rain is a fundamental element of weather observation and significantly affects many aspects of human life [1]. Rain impact extends across multiple sectors, including transportation [2], agriculture [3], public health [4], tourism [5], and finance [6], influencing daily activities as well as economic stability. A striking example concerning the rain effect was the 2015 floods in Jakarta, which submerged 20% of the city, evacuated 1,400 residents, and caused daily economic losses of approximately USD 114 million, according to the National Disaster Management Authority (BNPB) [7]. Despite the critical role of rainfall data in disaster management and planning, rain observation systems remain limited and insufficiently distributed [13]. There is a pressing need for cost-effective and easy-to-maintain rainfall intensity measurement systems that can be widely deployed across different locations.

Rainfall can be measured using primary or derivative methods, where the primary method includes using a rain gauge, which is simple to operate and maintain [8]. However, the method requires installation in unobstructed locations, which can limit its effectiveness in specific environments [9]. The derivative method, which includes radar and satellite-based measurements, has the advantage of covering large areas. However, this method is vulnerable to external interferences, such as signal disruptions from nearby fields, reflections from objects, and limitations in both temporal and spatial resolution [10]. High-resolution rainfall data, spatially and temporally, is crucial for hydrological and climate modeling. Accurate data improves the precision and reliability of simulations and predictions, making it an essential component in environmental and meteorological studies [11].

The need for high-resolution spatial and temporal rainfall data extends outside hydrological applications and plays a crucial role in the transportation sector, particularly in Intelligent Transport Systems (ITS) and human mobility management. Studies have consistently shown that weather conditions, specifically rainfall, significantly impact transportation [12], [13], often disrupting traffic flow and increasing congestion. Traffic congestion in highly urbanized areas such as Jakarta leads to considerable economic losses, estimated at $5 billion annually [14]. A practical method for mitigating congestion is the implementation of ITS, which provides real-time traffic updates, allowing drivers to select alternative routes and avoid heavily congested areas. However, the effectiveness of ITS depends on access to real-time weather data, particularly high-resolution rainfall intensity measurements, which are essential for predicting traffic behavior under varying weather conditions. Despite the importance of such data, the sparse distribution of rainfall observation instruments has created significant gaps, limiting the efficiency of ITS operations. To address the issue, crowd-sourced data from surveillance cameras presents a promising alternative. This camera, widely installed in urban areas, can estimate rainfall intensity, providing the high-resolution data needed to increase ITS performance and improve urban mobility management.

Recent advancements in deep learning have facilitated using a Convolutional Neural Network (CNN) for estimating rainfall intensity from surveillance camera images. Unlike traditional methods that rely on physical sensors such as rain gauges or radar, CNN-based models can automatically learn spatial and

temporal rainfall patterns directly from images. Several studies have shown the effectiveness of CNN in extracting rainfall-related features from image data, leading to improved estimation accuracy. For instance, Yin et al. [15], introduced an image-based deep-learning model to estimate urban rainfall intensity with high spatial and temporal resolution. The study used a modified Residual Network (ResNet)34 architecture, termed irCNN, which was trained on a dataset comprising both synthetic and real-time images captured from surveillance cameras and smartphones. The model achieved Mean Absolute Percentage Error (MAPE) between 16.5% and 21.9% in rainfall intensity estimation. Despite these promising results, challenges remain in ensuring the model's generalizability across different locations, camera types, and extreme weather conditions. Further validation and model improvements are necessary to increase strength and adaptability for real-world applications.

A two-stage framework has been developed for rainfall estimation, combining raindrop extraction with deep learning-based intensity estimation. In the first stage, raindrop extraction uses low-rank matrix decomposition [16], and Markov random fields [17]. In the second stage, rainfall intensity is estimated using a deep learning model, irCNN [18]. The dataset used for evaluation is collected from rainfall events in Hangzhou, China, and includes daytime and night-time conditions. During the process, ground truth data is obtained from a tipping-bucket rain gauge for accuracy assessment. The results show that preprocessing significantly improves performance in nighttime conditions, reducing MAPE to 19.73%. However, preprocessing slightly lowered accuracy for daytime images, with MAPE increasing from 17.06% (raw image) to 19.58%. The result signifies that it may not be necessary for daytime scenarios as preprocessing improves model strength in low-light conditions.

A deep learning-based method has been developed for estimating rainfall intensity using video footage from surveillance cameras. This method uses a Recurrent Neural Network (RNN), specifically Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) [19]. The results shows that GRU, optimized using Adam optimizer, achieved the lowest MAPE of 4.49%, while LSTM produced slightly higher errors with a MAPE of 5.67%. These signify that GRU is computationally efficient and maintains high rainfall estimation accuracy. Despite these promising results, the study has some distinguished limitations. The model is not tested on unseen image data from different cameras or locations, raising concerns about its ability to generalize to real-world conditions where environmental variables may differ significantly. Further validation is necessary to assess the strength and adaptability of the model across diverse settings.

Shalaby et al. [20], introduced a deep learning-based method for estimating rainfall intensity using video footage from surveillance cameras and smartphones. This method uses CNN to analyze rainfall patterns and consists of three primary stages: image preprocessing, CNN model training, and transfer learning. The model was initially trained on surveillance camera images and later fine-tuned with smartphone images to improve generalization. Relating to the discussion, the dataset collected at Monash University, Malaysia, between May and December 2022, included 6,121 images from surveillance cameras and 1,984 images from smartphones. The ground truth rainfall intensity measurements were obtained using a tipping-bucket rain gauge. The results showed that image preprocessing significantly improved model performance, with the best CNN model achieving an R^2 of 0.955 and Mean Absolute Error (MAE) of 2.508 mm/h for surveillance camera data.

Additionally, transfer learning improved prediction accuracy for smartphone images, producing R^2 of 0.840 and MAE of 4.374 mm/h. Despite the promising outcomes, the study had certain limitations. The model's generalization was restricted due to training on a single camera setup, and the study did not evaluate nighttime images, leaving its performance in low-light conditions uncertain. Further analysis is needed to improve strength across diverse camera environments and lighting conditions.

A study on rainfall intensity estimation using a surveillance camera with CNN has been widely conducted. However, two significant aspects require improvement. First, the accuracy of existing models remains a challenge. Despite previous studies showing that deep learning models successfully estimate rainfall, many methods rely on preprocessed inputs, such as rain streak removal or background subtraction, before feeding the data into CNN. Even though preprocessing improves feature extraction, it often introduces fixed assumptions about lighting conditions, scene background, and rainfall characteristics. This process can reduce adaptability to real-time variations such as changing illumination, environmental noise, and dynamic backgrounds. Second, many existing models struggle with generalization to unseen data. Several studies have reported that CNN-based rainfall estimation models perform well on training datasets but experience significant performance drops when tested on independent images from different locations, weather conditions, or camera configurations. This lack of generalization is partly due to dataset biases introduced by preprocessing methods that filter information essential for recognizing rainfall across diverse environments.

Addressing the complexities of real-world environments is crucial for effective image processing. This study introduces a groundbreaking method that eliminates the need for preprocessing by directly processing raw images. The method uses a strong CNN architecture to improve adaptability and simplify computational workflows. By performing this process, the analysis anticipates significant improvements in model accuracy and generalization capabilities, allowing seamless performance across diverse surveillance camera settings.

The primary objective of this study is to estimate rainfall intensity images using a Deep Learning model using ResNet architecture, leveraging the image captured by a surveillance camera. This study is driven by three major factors which include:

1) Toughness to temporal variability without preprocessing: Despite applying deep learning models to rainfall estimation, the ability to handle temporal variations remains underexplored. Environmental changes, such as fluctuations in lighting between day and night or seasonal variations, can significantly affect model performance. However, most existing models rely heavily on preprocessing methods to standardize image inputs, which can be

computationally expensive and impractical for real-time applications. This study aims to develop a ResNet-based model that inherently adapts to temporal variations, ensuring consistent and reliable rainfall estimation without extensive preprocessing.

2) Higher Temporal and Spatial Resolution Compared to Automatic Rain Gauge (ARG): Surveillance cameras offer higher temporal and spatial resolution than tipping bucket ARG, among the most widely used rainfall measurement systems. Relating to the discussion, a single surveillance camera can capture multiple images per second from various locations across a city, providing continuous and detailed data. This capability enables real-time monitoring and a more granular measurement of rainfall intensity, which is not achievable with the relatively sparse ARG network.

3) Limited Study on Models Evaluated with Unseen Image Data: A significant gap in the existing study is the lack of models tested on unseen images that are not part of the training process. Many existing models perform well on training and validation datasets but struggle to generalize when exposed to unseen data from the cameras used to train the model. Therefore, this study explicitly addresses the mentioned limitation by evaluating the model on entirely new image datasets, ensuring the toughness and adaptability of the model in real-world deployment.

The rest of this article is organized as follows: Section II presents related work, while Section III describes the methods. Section IV details the experimental setup, followed by Section V, which discusses the results. Finally, Section VI concludes the analysis and recommends directions for future study.

II. Related Works

Several studies have explored using surveillance cameras and CNN to estimate rainfall intensity. This method included analyzing images captured by CCTV cameras every 10 minutes at 85 locations during daylight hours (6:00 AM to 7:00 PM Singapore time). These images had resolutions of either 640 × 480 or 320 × 240 pixels [21]. Data from a tipping bucket rain gauge was recorded at 5-minute intervals to obtain rainfall labels. The Inverse Distance Weighting (IDW) [22] method was then applied to associate the rainfall labels with the corresponding camera locations. For rain removal, studies tested two deep learning models, namely VGG19 [23] and a hybrid method combining VGG19 with a Density-aware Image De- raining method using a Multi-stream Dense Network (DID-MDN) [24]. Both models incorporated three hidden layers (512, 256, 128), used ReLU activation, and were trained with ADAM optimizer-based gradient descent, set at a learning rate 0.25 [25]. The results showed that the hybrid model outperformed the standard VGG19 model. Additionally, high-quality images produced more accurate rainfall estimates compared to lower-resolution images.

Yin et al. [15], applied the irCNN model to estimate rainfall using CCTV images. The existing irCNN architecture was a modified version of ResNet [26], based explicitly on ResNet34, but with convolutional layers reduced to 29. The study used three types of data: synthetic rain image, image captured with a mobile phone camera, and image extracted from CCTV footage. Rainfall labels were obtained from rain gauge data recorded at one-minute intervals. However, since the temporal resolution of the gauge data did not match CCTV and mobile phone images, linear interpolation was applied to downscale the measurements accordingly. Yin initialized the irCNN model with pre-trained weights from ImageNet [27] for training. The results showed that the proposed model achieved MAPE ranging from 13.5% to 21.9%, signifying its effectiveness in estimating rainfall from image-based data sources.

During the analysis process, CNN combined with various RNN architectures, including SimpleRNN [28], LSTM [29], and Gated Recurrent Unit (GRU) [30], was used to estimate rainfall intensity from surveillance camera videos [19]. The study's CNN backbone is used for feature extraction as EfficientNetB0 [31]. To process the data, the original video recordings, captured at a resolution of 1920 × 1080 pixels, were cropped to 540 × 380 pixels. CCTV footage was acquired from a single fixed-location camera, while rainfall labels were obtained from ARG at one-minute intervals. The proposed model was evaluated using three optimization algorithms: Adam, RMSProp, and Stochastic Gradient Descent (SGD). The analysis showed that MAPE ranged from 3.55% to 6.95%, signifying the model's effectiveness in estimating rainfall from video-based data.

Zheng et al. [18], introduced a two-stage deep-learning framework designed to estimate rainfall intensity using video footage from a surveillance camera. The first stage focused on preprocessing and applied three primary methods, namely Low-Rank Matrix Decomposition (LRMD) [32] to separate the background from raindrop regions, Markov Random Fields (MRF) [33] for raindrop segmentation, and Sparse Optimization (SO) to improve raindrop visibility while minimizing noise. After preprocessing, the extracted rainfall features were fed into irCNN, a modified ResNet34 model version that continuously predicted rainfall intensity. Video recordings from 12 different rainfall events in Hangzhou, China, were used to evaluate the framework. These recordings captured daytime and night-time conditions, with ground truth intensity measured by a tipping-bucket rain gauge. Experimental results showed that preprocessing significantly improved nighttime performance, reducing MAPE to 19.73%. During the daytime, better accuracy was achieved when using raw images rather than preprocessed ones, with MAPE values of 17.06% compared to 19.58% after preprocessing. Despite the advancements of this model, the study has several limitations. Since the model was tested using data from a fixed camera setup, its ability to generalize to different camera sources or new locations remains uncertain. Another concern is that the preprocessing stage might remove subtle rainfall details, affecting estimation accuracy, particularly in high-intensity rainfall scenarios. The studies shows the importance of further validation to improve the model's reliability, specifically in extreme weather conditions and across diverse surveillance networks.

Even though deep learning and surveillance camera imagery have significantly improved rainfall estimation, several challenges remain unresolved. A significant issue is the reliance on extensive preprocessing methods, such as rain streak extraction and image decomposition, to improve model

accuracy. As these methods improve performance, the models add computational complexity, making real-time deployment difficult and reducing the ability to adapt to changing conditions such as lighting variations and seasonal shifts. Another limitation stems from the dependence on rain gauge data, which typically has a temporal resolution of one minute. Despite the accuracy, the sparse distribution of ARG restricts high-resolution spatial rainfall measurements. In areas without ARG coverage, interpolation methods must be used, which can introduce errors and reduce measurement reliability. Lastly, a critical challenge lies in deep learning models' strength when applied to images from diverse environmental settings. Many models perform well on the training datasets but struggle to generalize when faced with new camera locations, lighting conditions, or urban landscapes. This limitation reduces the effectiveness in real-world applications and shows the need for further improvements in model adaptability.

This study introduces a novel rainfall estimation model to improve accuracy, adaptability, and spatial resolution. The first significant improvement is the model's ability to handle temporal variations without relying on extensive preprocessing. By eliminating the need for computationally expensive preprocessing methods, the model can seamlessly adapt to changes in lighting and seasonal conditions, making real-time deployment more practical. Another significant advancement is the shift away from traditional rain gauge-based methods, which are constrained by the sparse distribution of ARG. Surveillance cameras, widely installed in urban areas, offer a significantly denser spatial network for rainfall monitoring. The proposed method can supplement ARG data by incorporating deep learning methods with camera imagery, providing a more detailed and continuous representation of rainfall intensity. It reduces dependence on interpolation methods and improves the accuracy of rainfall distribution mapping. Finally, it is tested on previously unseen datasets from different camera sources and locations to ensure the model's toughness across diverse environments. The evaluation process improves the ability of the model to generalize outside its training data, ensuring reliable performance in varying environmental conditions. The proposed method improves rainfall estimation accuracy, efficiency, and scalability by addressing these limitations using deep learning and surveillance camera imagery.

III. METHODS

VGG16 and ResNet34 architectures were previously used to estimate rainfall intensity from surveillance camera images or videos. However, the accuracy of the models remained a challenge and required further improvement. This limitation was a foundation for the decision to use the architectures and improve the performance by incorporating the Convolutional Block Attention Module (CBAM).

A. VGG16

VGG architecture was introduced by Simonyan [34] and featured 16 convolutional layers organized into five sequential blocks. The design incorporated increasing filters in deeper layers to improve feature extraction. Each convolutional layer used 3×3 kernels, ReLU activation, and the same padding. The first two blocks consisted of convolutional layers with 64 and 128 filters, followed by batch normalization and max pooling. Following the process, the third block contained three layers with 256 filters, while the fourth and fifth blocks each included three layers with 512 filters. VGG architecture has been widely applied in various fields, including rainfall intensity estimation [21], leaf disease classification [35], plant disease identification [36], and classification of individuals with dementia People [37].

B. ResNet34

ResNet was developed as CNN architecture to address performance degradation issues commonly observed in deep networks [26]. In traditional CNN architectures, each layer attempted to learn a direct mapping from input to output. However, as network depth increased, performance often deteriorated due to optimization challenges rather than overfitting. ResNet introduced residual learning to overcome this issue, allowing the model to learn a residual mapping rather than a direct one. The core component of ResNet was the residual block, which consisted of multiple nonlinear layers, including convolutional layers, batch normalization, and activation functions. These layers were combined with an identity shortcut connection that directly associated the input to the output of the block, helping to maintain gradient flow and improve training efficiency. ResNet has been widely applied in various fields, including colorectal cancer detection [38], pathologic myopia[39], and rainfall estimation [15].

C. Convolutional Block Attention Module (CBAM)

CBAM, introduced by Woo et al. [40], was developed as an attention mechanism designed to improve the performance of CNN. Designed as a lightweight and versatile component, CBAM could be seamlessly incorporated into any CNN architecture [40], [41].

The mechanism consisted of two major modules, the channel and spatial attention module, as shown in Fig. 1. The channel attention module identified the most important features by prioritizing relevant channels in the feature map. Consequently, the spatial attention module determined the locations of significant features, showing crucial spatial regions in the input data.

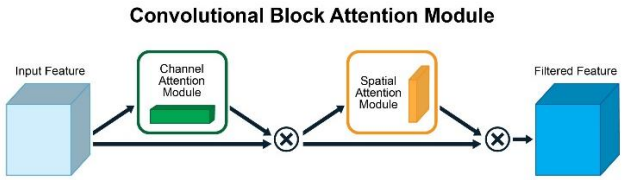

Fig. 1. Image of CBAM architecture [40].

When processing an intermediate feature map $F \in \mathbb{R}^{C \times H \times W}$ as input, CBAM first generated a 1D channel attention map $M_c \in \mathbb{R}^{C \times 1 \times 1}$ followed by 2D spatial attention map $M_s \in \mathbb{R}^{1 \times H \times W}$ as shown in Fig. 2 and Fig. 3. The complete attention mechanism operated through element-wise multiplication, represented by $\otimes$. During this operation, channel attention values were extended across the spatial dimensions, while spatial attention values were expanded along the channel dimension. The resulting feature map was represented after applying the channel attention module F'. The parameter F'' was the final output, as shown in Eq. (1) and Eq. (2). Fig. 2 and Fig. 3 further showed

the computation process of the channel and spatial attention module, respectively.

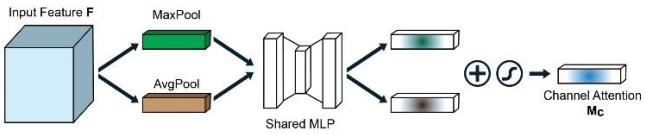

Fig. 2. Computational process for channel attention [40].

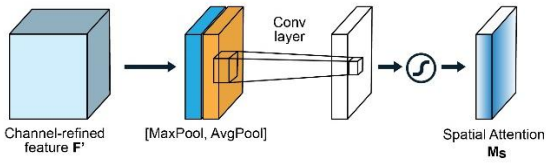

Fig. 3. Computational process for spatial attention [40].

$$F' = M_C(F) \otimes F \tag{1}$$

$$F'' = M_S(F') \otimes F' \tag{2}$$

The process began by capturing spatial information from the feature map through average and max pooling operations. These processes generated two distinct spatial context descriptors. F_C^{avg} representing the average-pooled features and $F_C^{\max}$ signifying the max pooling feature. Both descriptors were then processed through a shared network to compute the channel attention map $M_c \in \mathbb{R}^{C\times1\times1}$ where r was the reduction ratio. After passing each descriptor through the network, the resulting feature vectors were fused using element-wise summation. The channel attention was determined using the following computation in the following Eq. (3)

$$\begin{aligned} M_C(F) &= \sigma\left(MLP\left(AvgPool(F)\right) + MLP\left(MaxPool(F)\right)\right) \\ &= \sigma\left(W_1\left(W_o\left(F_{avg}^C\right)\right)\right) + \sigma\left(W_1\left(W_o\left(F_{\max}^C\right)\right)\right) \end{aligned} \tag{3}$$

where, σ represented the sigmoid function, while $W_0 \in \mathbb{R}^{C\times C/r}$ and $W_1 \in \mathbb{R}^{R/c\times C}$.

During this process, a spatial attention map was generated to capture the inter-spatial relationships of features. Unlike channel attention, which identified 'what' was important, spatial attention focused on 'where' the most informative regions were located, allowing it to be a complementary mechanism. The computation of spatial attention began with average and max pooling along the channel axis. These operations helped to form a compact yet practical feature descriptor by improving the visibility of crucial regions in the feature map [42]. The spatial attention map $M_s(F) \in \mathbb{R}^{HxW}$, was then generated by applying a convolution layer to the concatenated feature descriptor, enabling the model to signify or suppress specific regions. The process started with two pooling operations that aggregated channel information, producing two 2D feature maps, namely $F_s^{avg} \in \mathbb{R}^{1xHxW}$ and $F_s^{max} \in \mathbb{R}^{1xHxW}$, both computed across the channel dimension. These pooled feature maps were concatenated and processed through a standard convolution layer, producing the final 2D spatial attention map. In summation, the complete spatial attention mechanism was mathematically described in the following Eq. (4)

$$\begin{aligned} M_S(F) &= \sigma\left(f^{7\times7}\left(\left[AvgPool(F); MaxPool(F)\right]\right)\right) \\ &= \sigma\left(f^{7\times7}\left(\left[F_{avg}^s; F_{\max}^s\right]\right)\right) \end{aligned} \tag{4}$$

where, σ was the sigmoid function, and f^{7x7} referred to a convolution operation using a 7×7 filter size.

D. VGG-CBAM

VGG-CBAM refers to the combination of VGG architecture with the CBAM module in the context of this discussion. Previous studies investigated the application of VGG-CBAM, particularly in facial expression recognition [43] and image classification [41]. However, to the best available knowledge, no study has yet explored the use of the model for estimating rainfall intensity from CCTV footage.

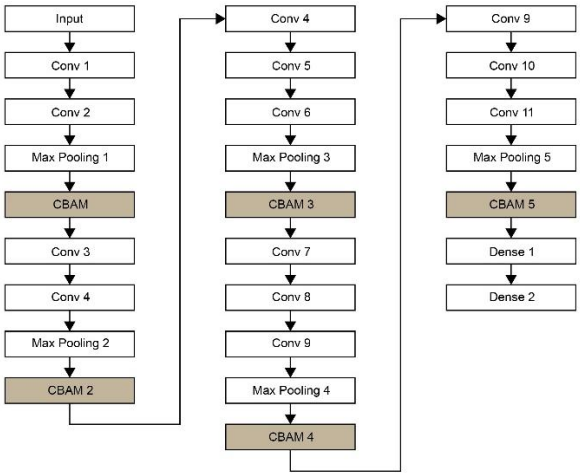

Fig. 4. Proposed architecture of VGG-CBAM model.

Fig. 4 shows VGG-CBAM architecture, which began with an input layer, followed by three convolutional layers (Conv 1, 2, and 3) to extract initial spatial features. After these layers, the Max Pooling 1 was applied to reduce spatial dimensions, followed by the first CBAM module (CBAM 1) to improve feature representation through spatial and channel-wise attention. The next stage consisted of three additional convolutional layers (Conv 4, 5, and 6), followed by Max Pooling 2 and CBAM 2 to refine the extracted features further. This pattern continued with another set of convolutional layers (Conv 7, 8, and 9), succeeded by Max Pooling 3 and CBAM 3. In the final segment, two more convolutional layers (Conv 10 and 11) were introduced, followed by Max Pooling 4 and the last attention module, CBAM 4. The network was completed with two dense layers responsible for aggregating extracted features and generating the final output. This design enabled the architecture to perform tasks such as classification or regression effectively. Several studies have applied VGG-CBAM in various fields, including bat classification [41] and facial expression recognition [43].

E. ResNet-CBAM

ResNet-CBAM refers to the integration of ResNet architecture with the CBAM module. Several studies have used this architecture for disease detection, including brain diseases [44] and lung cancer [45]. However, to the best of available knowledge, no studies have yet investigated the application of the model for estimating rainfall intensity.

Fig. 5 shows ResNet-CBAM architecture, as the design used a ResNet backbone configured with four stages of residual blocks, arranged into 3, 4, 6, and 3 blocks per stage, leading to 16 residual blocks across the network. Unlike traditional ResNet implementations, which typically included multiple pooling layers, this architecture used a single global average pooling layer immediately before the fully connected layer to reduce spatial dimensions. CBAM modules were strategically incorporated to improve feature representation, with 15 CBAM modules distributed across the residual blocks. These modules refined spatial and channel-wise feature maps, enabling the network to focus on the most relevant features. Several studies have applied ResNet-CBAM in different domains, including brain disease detection [44] and malignant–benign pulmonary nodule classification [45], multi-classification on arrhythmias [46], as well as flower classification [47].

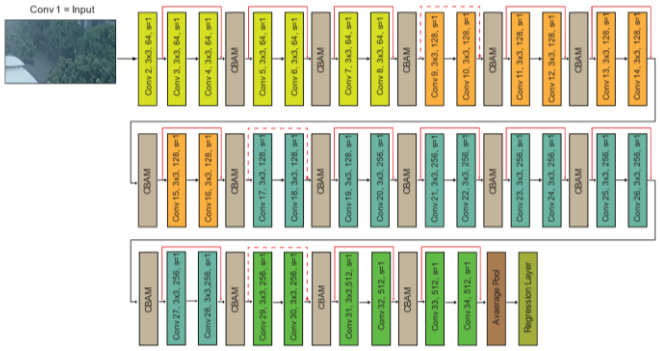

Fig. 5. The proposed architecture of the ResNet-CBAM model.

IV. Experimental Setup

A. Dataset

This study used Python 3.11 and TensorFlow 2.0, with the dataset stored in HDF5 files. The deep Learning model was executed on a PC equipped with an Intel(R) 11th Gen Core™ i7-11700F processor and NVIDIA RTX 3090 GPU with 32 GB of memory. Moreover, the dataset used in this study was captured with a surveillance camera featuring a resolution of 2560x1440 pixels and a frame rate of 25 frames per second (fps). Fig. 6 shows the samples of image rainfall captured during the study. The analysis focused on eight rainfall events recorded between January and April 2024.

Fig. 6. Image rainfall captured using a surveillance camera.

The selected rainfall events occurred at different times, including morning, afternoon, evening, and night. Data was collected during the analysis using ARG installed near the CCTV camera to ensure accurate rainfall measurements. The specific rainfall events included in the analysis are shown in Table I.

TABLE I. Rainfall Events Used for Dataset Creation

Rain event	Time event	Duration (minutes)	Period
January 29, 2024	06:26-06:43	17	Morning
February 06, 2024	06:51-07:19	28	Morning
February 04, 2024	11:55-13:55	120	Noon
March 14, 2024	11:54–12:31	37	Noon
March 31, 2024	15:15–15:47	32	Afternoon
April 09, 2024	16:05–17:40	95	Afternoon
January 27, 2024	00:10–00:54	44	Night
February 10, 2024	01:08–03:04	116	Night

This study used rainfall measurements from the tipping bucket type ARG as the ground truth dataset. The use of tipping bucket ARG for ground truth data has been extensively documented in previous studies [15], [19], [48], [49]. The device in this analysis operated with a resolution of 0.2 mm per minute. Additionally, the model's mechanism functioned by directing rainfall into a funnel, where it accumulated in a small bucket. After reaching full capacity, the bucket tipped, registering a measurement of 0.2 mm of rainfall. However, this design imposed a limitation, as the model could not record rainfall intensities below 0.2 mm, even when rainfall was visibly present in camera imagery.

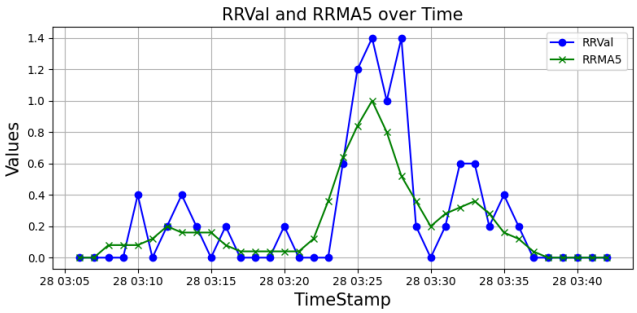

Fig. 7. Comparison between rainfall real value and rainfall after moving average five window.

Fig. 7 showed that the raw rainfall data collected from ARG indicated significant variability, with unexpected increase in rainfall followed by abrupt decrease. This pattern signified that per-minute rainfall measurements from ARG might not accurately reflect actual rainfall intensity [50]. Previous studies recommended an appropriate temporal resolution for ARG data ranging between 5 and 10 minutes [51]. A five-minute moving average method was applied to address the high fluctuations in the data. This method computed the average rainfall intensity over a rolling five-minute window, producing a smoother dataset that reduced the impact of rapid spikes or drops observed in minute-by-minute measurements. The mathematical

formulation of the moving average method was presented in the following Eq. (5)

$$R_t = \frac{R_{t-2} + R_{t-1} + R_t + R_{t-+} + R_{t+2}}{5} \tag{5}$$

In the moving average formula, R_t represented the rainfall intensity at time t. Similar to the previous value, R_{t-1} and R_{t-2} corresponded to the rainfall intensities recorded 1 minute and 2 minutes before t, respectively. R_{t+1} and R_{t+2} represented the rainfall intensities, which recorded 1 minute and 2 minutes after t, while R_0 signified the current rainfall concentration. After the data had been smoothed using the moving average method, the next step included applying linear interpolation to the per-second data. This step was crucial for ensuring precise rainfall intensity values at each second, enabling synchronization with the image captured by the surveillance camera. The formula for linear interpolation was presented in the following Eq. (6)

$$I_t = I_L + \frac{t}{60}\left(I_R + I_L\right) \tag{6}$$

where, it represented the rainfall intensity at second t, I_L was the rainfall concentration at the start of the minute, I_R signified the intensity at the last minute, and t indicated the specific second being evaluated.

Fig. 8 shows the results of rainfall observations and interpolation during the analysis. The black line represented observed rainfall intensity, while the red line signified the results of linear interpolation. The image captured by the camera during the process had a resolution of 2660×1440 pixels. However, full resolution was not used as input; instead, a random selection of 180×120 pixels was performed for the processing. The dataset used during the analysis comprised 49,005 samples divided into three subsets, including 70% used for training, 15% for validation, and 15% for testing, respectively.

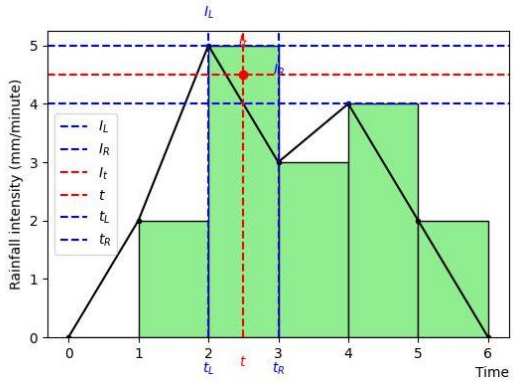

Fig. 8. Rainfall intensity interpolation over time.

B. Evaluation Criteria

The following formulas were used to assess regression model performance, including MAE, Mean Arctangent Absolute Percentage Error (MAAPE), Nash-Sutcliffe Efficiency (NSE), Kling-Gupta Efficiency (KGE), and the Coefficient of Determination (R^2).

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|Y_i - \hat{Y}_i\right| \tag{7}$$

$$MAAPE = \frac{1}{n}\sum_{i=1}^{n}\arctan\left|\frac{Y_i - \hat{Y}_i}{Y_i}\right| \tag{8}$$

$$NSE = 1 - \frac{1}{n}\frac{\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2}{\left(Y_i - \bar{Y}_i\right)^2} \tag{9}$$

$$R^2 = \frac{\left(\sum_{i=1}^{n}\left(Y_i - \tilde{Y}\right)\sum_{i=1}^{n}\left(Y_i - \hat{Y}\right)\right)^2}{\sum_{i=1}^{n}\left(Y_i - \tilde{Y}\right)^2\sum_{i=1}^{n}\left(Y_i - \hat{Y}\right)^2} \tag{10}$$

$$KGE = 1 - \sqrt{(r-1)^2 + \left(\frac{\sigma_{pred}}{\sigma_{obs}} - 1\right)^2 + \left(\frac{\mu_{pred}}{\mu_{obs}} - 1\right)^2} \tag{11}$$

V. Result

Four models were evaluated in this study, namely VGG16, ResNet34, and their respective versions integrated with CBAM. The selection of VGG16 and ResNet34 was based on proven effectiveness in previous studies on rainfall intensity estimation using surveillance camera imagery.

Table II shows the evaluation metrics for four models, including VGG16, ResNet34, VGG16-CBAM, and ResNet34-CBAM.

TABLE II. Performance Metrics of Evaluated Models in Testing Phase

Model	MAAPE	MAE	NSE	KGE	RMSE	R^2
VGG16	10.06	0.04	0.94	0.95	0.08	0.94
ResNet34	14.34	0.06	0.91	0.94	0.10	0.91
VGG16-CBAM	9.3	0.03	0.95	0.96	0.07	0.95
ResNet34-CBAM	13.42	0.05	0.92	0.95	0.09	0.92

A comparison between the original VGG16 and ResNet32 models showed significant differences in the performance for rainfall estimation. VGG16 consistently outperformed ResNet34 across all evaluation metrics, signifying a stronger ability to capture complex patterns in the data. This advantage was attributed to the deeper and more feature-rich architecture of VGG16, which facilitated more effective feature extraction and processing. ResNet34, despite using residual connections to mitigate vanishing gradient issues, failed to achieve comparable results. The outcome recommended that the architectural design of the model was less suited for this specific task. Further analysis comparing VGG16 to its improved version, VGG16-CBAM, showed the impact of incorporating attention mechanisms. The addition of CBAM enabled the model to prioritize critical spatial and channel-wise features, improving accuracy and error reduction across all evaluation metrics. These

results signified the effectiveness of attention mechanisms in refining feature extraction and improving the capacity of the model to detect relevant patterns in complex datasets. Finally, VGG16-CBAM surfaced as a more effective model than the original VGG16.

When evaluating ResNet34 and ResNet34-CBAM models, the incorporation of CBAM showed a clear performance improvement. By enabling the model to signify important features while suppressing irrelevant factors selectively, the CBAM module contributed to higher predictive accuracy and reduced errors. Despite these improvements, ResNet34-CBAM still underperformed compared to both VGG16 and VGG16-CBAM. This result implied that while attention mechanisms offered benefits, the structural limitations of ResNet34 constrained the total effectiveness of the model for rainfall estimation. A direct comparison between VGG16-CBAM and ResNet32-CBAM further reinforced the superiority of VGG16 architecture. Relating to this discussion, VGG16-CBAM consistently delivered better results across all evaluation metrics, showing that combining a strong base architecture with an advanced attention mechanism produced the most effective model. These results signified that although attention mechanisms such as CBAM improved performance across different architectures, the total effectiveness still heavily depended on the fundamental design of the base model. It showed the importance of carefully selecting and optimizing model architectures for superior performance in complex data modeling tasks.

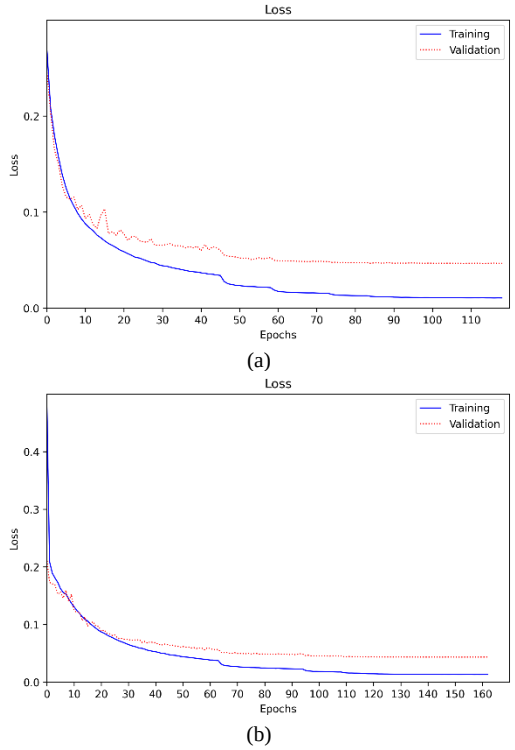

Fig. 9. Training and validation loss curve. (a) VGG16, (b) VGG16-CBAM.

Fig. 9 shows the loss graphs for VGG16 and VGG16-CBAM, indicating significant differences in training and validation performance. For the VGG16 model, the training loss gradually decreased with the number of epochs, eventually converging to a low value. However, a significant gap existed between the training and validation losses, with the validation loss remaining consistently higher. The discrepancy showed that despite effectively learning patterns of the VGG16 model from the training data, generalization to unseen validation data was limited, signifying potential overfitting. Consequently, the VGG16-CBAM model signified improved performance, as reflected in its loss curves. Both training and validation losses decreased gradually and converged to lower values than those observed in the VGG16 model. Moreover, the validation loss closely followed the training loss in the epochs, indicating improved generalization. The reduced gap between training and validation losses showed that the CBAM module enabled the model to focus on relevant features more effectively, mitigating overfitting and improving performance on unseen.

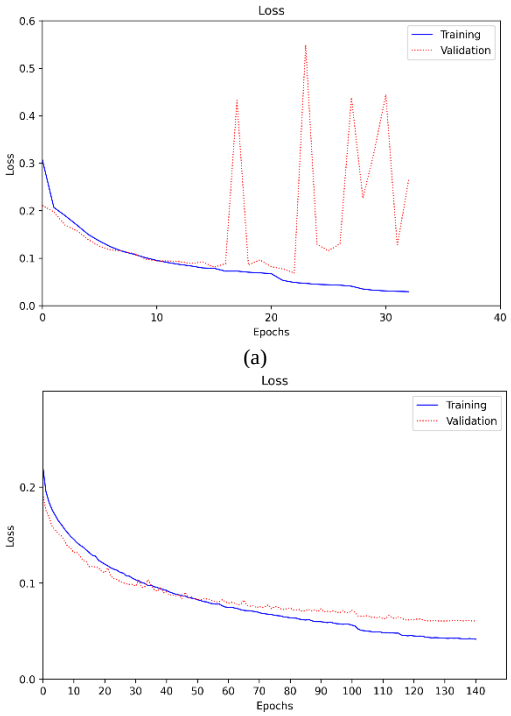

Fig. 10. Training and validation loss curve of (a) ResNet34 and (b) ResNet34-CBAM.

Fig. 10 compared the training and validation loss of ResNet34 and ResNet34 incorporated with CBAM, showing key differences in the learning behavior. For the ResNet34 model, training loss gradually decreased, signifying effective learning. However, the validation loss showed significant fluctuations, specifically in the early training phase, implying instability and potential overfitting. Due to this instability, early stopping was triggered sooner, as the validation loss failed to show consistent improvement, reflecting the limited generalization capability of the model. Consequently, the ResNet34-CBAM model signified a smoother and more stable decline in training and validation loss. The validation loss closely followed the training loss in training, indicating improved generalization and reduced overfitting. Early stopping occurred later for ResNet34-CBAM as the model improved its

validation performance. The inclusion of CBAM improved the ability of the model to focus on relevant features, leading to superior stability, lower overall loss, and better performance. These results showed ResNet34-CBAM as a more effective architecture for this task.

Fig. 9(b) and Fig. 10(b) presented a comparison of the training and validation loss for VGG-CBAM and ResNet-CBAM, respectively, in the training process. In Fig. 9, which showed VGG-CBAM, the validation loss consistently decreased with minimal fluctuations and eventually reached a lower value than ResNet-CBAM. Additionally, the difference between training and validation loss in VGG-CBAM remained relatively small, indicating that the model generalized well and maintained stable training performance. Fig. 10, which showed ResNet-CBAM, signified a slightly slower decline in validation loss, with the final loss values remaining slightly higher than those observed in VGG-CBAM. Although ResNet-CBAM showed stable and consistent learning, its ability to minimize validation loss was marginally weaker than VGG-CBAM. The results signified that while both models benefited from integrating the CBAM module, VGG-CBAM achieved better final validation loss and convergence efficiency.

Table III shows the performance metrics of the proposed models, VGG-CBAM (Model I) and ResNet-CBAM (Model II), across different periods, namely morning, noon, afternoon, and night. The evaluation used images captured by an unseen camera that had not been included in the training phase. The results showed that both models maintained strong predictive capabilities even when tested on previously unseen data, signifying the ability to generalize effectively in rainfall estimation tasks. Although both models showed strong predictive capabilities across all periods, slight variations in performance appeared based on the time of day. Specifically, the morning and noon intervals showed slightly lower NSE and R^2 scores compared to the nighttime period. This discrepancy could be attributed to increased environmental noise, fluctuations in illumination, or atmospheric disturbances, all of which potentially affected the quality of image captured by the unseen camera.

TABLE III. PERFORMANCE EVALUATION OF THE PROPOSED MODELS ON UNSEEN IMAGES

Time	Model	MAAPE	MAE	NSE	KGE	RMSE	R^2
Morning	I	0.19	0.05	0.80	0.83	0.05	0.80
	II	0.42	0.05	0.83	0.85	0.06	0.83
Noon	I	0.42	0.05	0.82	0.78	0.06	0.82
	II	0.43	0.04	0.77	0.71	0.05	0.77
Afternoon	I	0.45	0.05	0.82	0.75	0.06	0.82
	II	0.80	0.05	0.76	0.84	0.05	0.76
Night	I	0.11	0.06	0.98	0.93	0.07	0.98
	II	0.1	0.05	0.98	0.94	0.06	0.98

During the afternoon, a noticeable increase in MAAPE was observed, particularly for ResNet-CBAM (Model II). It showed that fluctuating daylight conditions introduced additional challenges in feature extraction and rainfall estimation accuracy. The presence of dynamic shadows, variable cloud coverage, and rapid changes in lighting intensity probably affected the model's ability to consistently recognize rainfall-related patterns in surveillance images. Despite these variations, both models maintained highly reliable performance, as reflected in the consistently strong NSE, R^2, and KGE scores across different periods. These results showed the adaptability of the models to diverse lighting and environmental conditions, signifying the suitability for real-world applications where data acquisition was subjected to temporal variability.

VI. CONCLUSION AND FUTURE WORK

In conclusion, this study developed two improved deep learning models, VGG-CBAM and ResNet-CBAM, for estimating rainfall intensity using surveillance camera images. The VGG-CBAM model combined the tough feature extraction capabilities of VGG16 with CBAM modules, while the ResNet-CBAM model incorporated CBAM into ResNet34 architecture, which consisted of 16 residual blocks. Both models outperformed the baseline counterparts, with VGG-CBAM signifying the highest accuracy and toughness across various evaluation metrics. Incorporating CBAM modules improved spatial and channel-wise attention, allowing the models to capture fine-grained rainfall patterns more effectively. Surveillance cameras as a data source offered a scalable and cost-effective alternative to traditional rainfall observation methods. Relating to this discussion, future studies would focus on optimizing the architecture, incorporating transfer learning, and expanding the dataset to improve the generalizability of the models across diverse urban environments and weather conditions.

ACKNOWLEDGMENT

This study was conducted as part of a collaboration between Pusat Pengembangan Sumber Daya Manusia (PPDSM) and Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Indonesia (FMIPA UI), under a formal cooperation agreement. This agreement is documented as PKS HK.08.00/001/PPK/PDL/II/2024, issued by BMKG, and 249/PKS/FMIPA/UI/2024, issued by Universitas Indonesia. This study was also supported by PUTI Hibah 2023.

REFERENCES

[1] Q. Sun, C. Miao, Q. Duan, H. Ashouri, S. Sorooshian, and K. L. Hsu, "A Review of Global Precipitation Data Sets: Data Sources, Estimation, and Intercomparisons," Reviews of Geophysics, vol. 56, no. 1, pp. 79–107, Mar. 2018, doi: 10.1002/2017RG000574.

[2] M. Vidas, V. Tubić, I. Ivanović, and M. Subotić, "One Approach to Quantifying Rainfall Impact on the Traffic Flow of a Specific Freeway Segment," Sustainability (Switzerland), vol. 14, no. 9, May 2022, doi: 10.3390/su14094985.

[3] B. O. Olivares, F. Paredes, J. C. Rey, D. Lobo, and S. Galvis-Causil, "The relationship between the normalized difference vegetation index, rainfall, and potential evapotranspiration in a banana plantation of Venezuela," Social Psychology and Society, vol. 12, no. 2, pp. 58–64, 2021, doi: 10.20961/STJSSA.V18I1.50379.

[4] N. A. M. Salim et al., "Prediction of dengue outbreak in Selangor Malaysia using machine learning techniques," Sci Rep, vol. 11, no. 1, Dec. 2021, doi: 10.1038/s41598-020-79193-2.

[5] S. Franzoni and C. Pelizzari, "Rainfall financial risk assessment in the hospitality industry," International Journal of Contemporary Hospitality Management, vol. 31, no. 3, pp. 1104–1121, Apr. 2019, doi: 10.1108/IJCHM-10-2017-0632.

[6] G. S. Araujo, W. Mendes-Da-, S. Fundação, and G. Vargas, "Does Extreme Rainfall Lead to Heavy Economic Losses in the Food Industry? Open Science View project Market Risk View project", doi: 10.13140/RG.2.2.12822.24645.

[7] Siswanto et al., "A very unusual precipitation event associated with the 2015 floods in Jakarta: an analysis of the meteorological factors," Weather Clim Extrem, vol. 16, pp. 23–28, Jun. 2017, doi: 10.1016/j.wace.2017.03.003.

[8] J. A. Prakosa, S. Wijonarko, and D. Rustandi, "The performance measurement test on rain gauge of tipping bucket due to controlling of the water flow rate," in Proceedings of the 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering, ElConRus 2018, Institute of Electrical and Electronics Engineers Inc., Mar. 2018, pp. 1136–1140. doi: 10.1109/EIConRus.2018.8317291.

[9] C. W. Lin, M. Lin, and S. Yang, "SOPNET method for the fine-grained measurement and prediction of precipitation intensity using outdoor surveillance cameras," IEEE Access, vol. 8, pp. 188813–188824, 2020, doi: 10.1109/ACCESS.2020.3032430.

[10] C. W. Lin, X. Huang, M. Lin, and S. Hong, "SF-CNN: Signal Filtering Convolutional Neural Network for Precipitation Intensity Estimation," Sensors, vol. 22, no. 2, Jan. 2022, doi: 10.3390/s22020551.

[11] Z. Zhou, D. Lu, B. Yong, Z. Shen, H. Wu, and L. Yu, "Evaluation of GPM-IMERG Precipitation Product at Multiple Spatial and Sub-Daily Temporal Scales over Mainland China," Remote Sens (Basel), vol. 15, no. 5, Mar. 2023, doi: 10.3390/rs15051237.

[12] A. Theofilatos and G. Yannis, "A review of the effect of traffic and weather characteristics on road safety," Accid Anal Prev, vol. 72, pp. 244–256, 2014, doi: 10.1016/j.aap.2014.06.017.

[13] A. Romanowska and M. Budzyński, "Investigating the Impact of Weather Conditions and Time of Day on Traffic Flow Characteristics," Weather, Climate, and Society, vol. 14, no. 3, pp. 823–833, Jul. 2022, doi: 10.1175/WCAS-D-22-0012.1.

[14] C. L. Yang, H. Sutrisno, A. S. Chan, H. Tampubolon, and B. S. Wibowo, "Identification and analysis of weather-sensitive roads based on smartphone sensor data: A case study in Jakarta," Sensors, vol. 21, no. 7, Apr. 2021, doi: 10.3390/s21072405.

[15] H. Yin, F. Zheng, H. F. Duan, D. Savic, and Z. Kapelan, "Estimating Rainfall Intensity Using an Image-Based Deep Learning Model," Engineering, vol. 21, pp. 162–174, Feb. 2023, doi: 10.1016/j.eng.2021.11.021.

[16] M. Li et al., "Video Rain Streak Removal by Multiscale Convolutional Sparse Coding," in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 6644–6653. doi: 10.1109/CVPR.2018.00695.

[17] W. Wei, L. Yi, Q. Xie, Q. Zhao, D. Meng, and Z. Xu, "Should We Encode Rain Streaks in Video as Deterministic or Stochastic?," in 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2535–2544. doi: 10.1109/ICCV.2017.275.

[18] F. Zheng et al., "Toward Improved Real-Time Rainfall Intensity Estimation Using Video Surveillance Cameras," Water Resour Res, vol. 59, no. 8, Aug. 2023, doi: 10.1029/2023WR034831.

[19] F. Rajabi, N. Faraji, and M. Hashemi, "An efficient video-based rainfall intensity estimation employing different recurrent neural network models," Earth Sci Inform, 2024, doi: 10.1007/s12145-024-01290-x.

[20] Y. Shalaby, M. I. I. Alkhatib, A. Talei, T. K. Chang, M. F. Chow, and V. R. N. Pauwels, "Estimating Rainfall Intensity Using an Image-Based Convolutional Neural Network Inversion Technique for Potential Crowd-sourcing Applications in Urban Areas," Big Data and Cognitive Computing, vol. 8, no. 10, Oct. 2024, doi: 10.3390/bdcc8100126.

[21] R. Zen, D. M. S. Arsa, R. Zhang, N. A. S. Er, and S. Bressan, "Rainfall Estimation from Traffic Cameras," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Springer, 2019, pp. 18–32. doi: 10.1007/978-3-030-27615-7_2.

[22] F. W. Chen and C. W. Liu, "Estimation of the spatial rainfall distribution using inverse distance weighting (IDW) in the middle of Taiwan," Paddy and Water Environment, vol. 10, no. 3, pp. 209–222, Sep. 2012, doi: 10.1007/s10333-012-0319-1.

[23] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," Sep. 2014, [Online]. Available: http://arxiv.org/abs/1409.1556

[24] H. Zhang and V. M. Patel, "Density-Aware Single Image De-raining Using a Multi-stream Dense Network," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, Dec. 2018, pp. 695–704. doi: 10.1109/CVPR.2018.00079.

[25] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," Dec. 2014, [Online]. Available: http://arxiv.org/abs/1412.6980

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition." [Online]. Available: http://image-net.org/challenges/LSVRC/2015/

[27] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255. doi: 10.1109/CVPR.2009.5206848.

[28] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A Critical Review of Recurrent Neural Networks for Sequence Learning," May 2015, [Online]. Available: http://arxiv.org/abs/1506.00019

[29] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Comput, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[30] C. Gulcehre, K. Cho, R. Pascanu, and Y. Bengio, "Learned-Norm Pooling for Deep Feedforward and Recurrent Neural Networks," Nov. 2013, [Online]. Available: http://arxiv.org/abs/1311.1780

[31] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," May 2019, [Online]. Available: http://arxiv.org/abs/1905.11946

[32] M. Li et al., "Video Rain Streak Removal By Multiscale Convolutional Sparse Coding."

[33] W. Wei, L. Yi, Q. Xie, Q. Zhao, D. Meng, and Z. Xu, "Should We Encode Rain Streaks in Video as Deterministic or Stochastic?"

[34] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," Sep. 2014, [Online]. Available: http://arxiv.org/abs/1409.1556

[35] K. Wei et al., "Explainable Deep Learning Study for Leaf Disease Classification," Agronomy, vol. 12, no. 5, May 2022, doi: 10.3390/agronomy12051035.

[36] J. Chen, J. Chen, D. Zhang, Y. Sun, and Y. A. Nanehkaran, "Using deep transfer learning for image-based plant disease identification," Comput Electron Agric, vol. 173, Jun. 2020, doi: 10.1016/j.compag.2020.105393.

[37] A. Bagaskara and M. Suryanegara, "Evaluation of VGG-16 and VGG-19 Deep Learning Architecture for Classifying Dementia People," in 2021 4th International Conference of Computer and Informatics Engineering

(IC2IE), IEEE, Sep. 2021, pp. 1–4. doi: 10.1109/IC2IE53219.2021.9649132.

[38] D. Sarwinda, R. H. Paradisa, A. Bustamam, and P. Anggia, "Deep Learning in Image Classification using Residual Network (ResNet) Variants for Detection of Colorectal Cancer," in Procedia Computer Science, Elsevier B.V., 2021, pp. 423–431. doi: 10.1016/j.procs.2021.01.025.

[39] Z. R. Himami, A. Bustamam, and P. Anki, "Deep Learning in Image Classification using Dense Networks and Residual Networks for Pathologic Myopia Detection," in 2021 International Conference on Artificial Intelligence and Big Data Analytics, ICAIBDA 2021, Institute of Electrical and Electronics Engineers Inc., 2021, pp. 191–196. doi: 10.1109/ICAIBDA53487.2021.9689744.

[40] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional Block Attention Module," Jul. 2018, [Online]. Available: http://arxiv.org/abs/1807.06521

[41] Z. Cao et al., "Fine-grained image classification on bats using VGG16-CBAM: a practical example with 7 horseshoe bats taxa (CHIROPTERA: Rhinolophidae: Rhinolophus) from Southern China," Front Zool, vol. 21, no. 1, Dec. 2024, doi: 10.1186/s12983-024-00531-5.

[42] S. Zagoruyko and N. Komodakis, "Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer," Dec. 2016, [Online]. Available: http://arxiv.org/abs/1612.03928

[43] W. Cao, Z. Feng, D. Zhang, and Y. Huang, "Facial Expression Recognition via a CBAM Embedded Network," in Procedia Computer Science, Elsevier B.V., 2020, pp. 463–477. doi: 10.1016/j.procs.2020.06.115.

[44] Y. Xiao, H. Yin, S. H. Wang, and Y. D. Zhang, "TReC: Transferred ResNet and CBAM for Detecting Brain Diseases," Front Neuroinform, vol. 15, Dec. 2021, doi: 10.3389/fninf.2021.781551.

[45] Y. Zhang et al., "Deep-Learning Model of ResNet Combined with CBAM for Malignant–Benign Pulmonary Nodules Classification on Computed Tomography Images," Medicina (Lithuania), vol. 59, no. 6, Jun. 2023, doi: 10.3390/medicina59061088.

[46] K. Ma, C. A. Zhan, and F. Yang, "Multi-classification of arrhythmias using ResNet with CBAM on CWGAN-GP augmented ECG Gramian Angular Summation Field," Biomed Signal Process Control, vol. 77, Aug. 2022, doi: 10.1016/j.bspc.2022.103684.

[47] K. Il Bae, J. Park, J. Lee, Y. Lee, and C. Lim, "Flower classification with modified multimodal convolutional neural networks," Expert Syst Appl, vol. 159, Nov. 2020, doi: 10.1016/j.eswa.2020.113455.

[48] J. B. Haurum, C. H. Bahnsen, and T. B. Moeslund, "Is it Raining Outside? Detection of Rainfall using General-Purpose Surveillance Cameras," Aug. 2019, doi: 10.5281/zenodo.4715681.

[49] S. Jiang, V. Babovic, Y. Zheng, and J. Xiong, "Advancing Opportunistic Sensing in Hydrology: A Novel Approach to Measuring Rainfall With Ordinary Surveillance Cameras," Water Resour Res, vol. 55, no. 4, pp. 3004–3027, Apr. 2019, doi: 10.1029/2018WR024480.

[50] D. Dunkerley, "Acquiring unbiased rainfall duration and intensity data from tipping-bucket rain gauges: A new approach using synchronised acoustic recordings," Atmos Res, vol. 244, Nov. 2020, doi: 10.1016/j.atmosres.2020.105055.

[51] E. Habib, W. F. Krajewski, and A. Kruger, "Sampling Errors of Tipping-Bucket Rain Gauge Measurements," J Hydrol Eng, vol. 6, no. 2, pp. 159–166, Apr. 2001, doi: 10.1061/(asce)1084-0699(2001)6:2(159).