Towards a Safer Construction Environment: Evaluating a Simple CNN for Safety Classification

Darius Vincent C. Ardales
College of Computer Studies, De La Salle University Manila, Philippines
darius_ardales@dlsu.edu.ph

***Abstract*– The construction industry faces significant safety challenges due to inconsistent use of personal protective equipment (PPE), leading to numerous injuries and fatalities annually. This paper presents a simple convolutional neural network (SafetyCNN) designed to classify construction site images as "safe" or "unsafe" based on PPE compliance. Leveraging data augmentation techniques and Test-Time Augmentation (TTA), the model enhances generalization and robustness against image variability. Emphasizing recall to minimize false negatives, the SafetyCNN achieved a recall of 93.55% on the training set and 72.46% on the test set. Analysis of the convolutional layers revealed effective learning of critical visual patterns associated with PPE usage. The results demonstrate that a straightforward CNN architecture, combined with effective augmentation strategies, can be a cost-effective and scalable solution for automated safety monitoring in construction environments.**

Keywords—Digital Image Processing, Convolutional Neural Networks, Data Augmentation, Affine Transformations, Test-time Augmentation, Worksite Safety

I. INTRODUCTION

The construction industry is fraught with hazards, leading to significant numbers of workplace injuries and fatalities annually. In 2020, occupational struck-by incidents alone resulted in 150 deaths and 14,000 nonfatal injuries in the construction sector, amounting to $1.4 billion in direct workers' compensation costs for nonfatal claims requiring more than five days away from work [1]. Additionally, falls, slips, and trips accounted for over 46% of fatal injuries in construction in 2021, highlighting the critical importance of safety interventions [2]. Despite advancements in safety protocols, implementing personal protective equipment (PPE), such as helmets and reflective vests, remains inconsistent, posing significant risks to workers.

Improving safety measures through automated systems can mitigate these risks. Recent studies emphasize the importance of incorporating advanced technology in safety management. For example, a study explored construction site understanding to enhance monitoring systems, demonstrating that machine learning algorithms significantly contribute to hazard detection and prevention [3]. However, current systems often employ complex deep-learning models that require substantial computational resources, which may not always be feasible in practical scenarios.

Previous studies have developed sophisticated models for PPE detection. For instance, YOLOv7 was used to detect safety equipment, achieving a high mean average precision of 87.7% [4]. Similarly, a lightweight CNN architecture was proposed, optimized for helmet detection with improved generalization capabilities [5]. These models demonstrate the potential of automated systems but often rely on complex architectures or external datasets, creating barriers to scalability and deployment in resource-constrained environments. In contrast, this study focuses on leveraging a simple CNN architecture to classify construction images while maintaining robust performance.

A. Objectives

This paper aims to:

1. Utilize digital image processing techniques, including affine transformations and color manipulation, for data augmentation.
2. Design and implement a straightforward CNN model for binary classification of construction site images as "safe" or "unsafe" while focusing on improving recall.
3. Analyze the learned features of the CNN model to understand its decision-making process, focusing on identifying critical visual patterns.

B. Model Overview

The proposed CNN model categorizes images into two classes:

1. **Safe:** Images where workers wear essential PPE (helmets and reflective vests).
2. **Unsafe:** Images lacking proper PPE or showcasing non-compliance.

The model employs basic convolutional layers and leverages data augmentation strategies to enhance generalization. By focusing on simplicity, this approach ensures computational efficiency while maintaining a high recall, minimizing the risk of misclassifying unsafe scenarios.

C. Significance of the Research

The study contributes to the field by demonstrating that a basic CNN architecture can effectively classify

construction site images, paving the way for cost-effective and scalable safety monitoring solutions.

D. Scope and Limitations

The study is limited to binary classification using a Kaggle dataset comprising images depicting construction workers with or without PPE.

Limitations:

1. The dataset may not fully represent real-world conditions, such as varying lighting or obstructions.
2. Time constraints prevented extensive hyperparameter tuning or exploration of alternative architectures.
3. The focus is on static image classification, excluding dynamic scenarios or real-time applications.

II. METHODOLOGY

A. Dataset Description

This study utilizes a publicly available dataset from Kaggle, curated explicitly for training and evaluating deep learning models concerning workplace safety compliance on construction sites. The dataset is tailored for binary image classification tasks, focusing on identifying whether images meet safety standards based on the presence of Personal Protective Equipment (PPE). It comprises 2,020 640x640 images, equally distributed into two categories: **safe** (1,010 images) and **unsafe** (1,010 images) [6]. The dataset was designed with high reliability, as a research team manually labeled images to ensure alignment with established safety standards. According to the dataset's author, the images were collected from various online sources and did not include bounding box annotations. Instead, the dataset strictly supports classification tasks, making it ideal for testing models that evaluate overall compliance in workplace imagery.

To determine whether an image is classified as "safe," all individuals depicted must wear a construction vest (commonly orange or blue with reflective strips) and a safety helmet. Any deviation, such as one or more individuals missing the required PPE, results in the image being categorized as "unsafe." This binary classification approach ensures that the model is sensitive to the most minor lapses in compliance, which is critical for real-world applications in safety monitoring.

B. Tech Stack

The study employs various tools and frameworks to facilitate image processing, model development, evaluation, and deployment. OpenCV is used for image preprocessing, while PyTorch is the primary framework for model creation, training, and analysis. Metrics such as accuracy, precision, recall, and F1-score are computed using scikit-learn. FastAPI is leveraged to deploy the trained model, enabling integration with platforms like HuggingFace Spaces for interactive usage. Other essential libraries include NumPy, Pandas, matplotlib, and TQDM for data manipulation, visualization, and process tracking. The implementation ensures reproducibility by setting consistent random seeds, and the computations are optimized for GPU where available.

C. Data Preparation and Preprocessing

The dataset underwent several preprocessing steps to ensure consistency and compatibility with the model. All images, originally sized at 640x640, were transformed using PyTorch's Torchvision transforms to maintain uniformity. The preprocessing pipeline included resizing to 640x640 pixels, conversion to tensors normalized between 0 and 1, and further normalization to a range of -1 to 1 using a mean and standard deviation of (0.5, 0.5, 0.5). After that, the dataset was loaded using ImageFolder, with folder names serving as labels ('safe' mapped to 0 and 'unsafe' mapped to 1). This structure streamlined the loading process and ensured proper label assignment.

A train-test split was performed, allocating 80% of the data for training (1,616 images) and 20% for testing (404 images), using random_split to maintain class distribution.

D. Train Data Augmentation

To enhance the generalizability of the model, the training data was augmented using five techniques:

1. Randomly placing a black patch.

Fig 1. Image with Black Patch

2. Shifting an image horizontally or vertically.

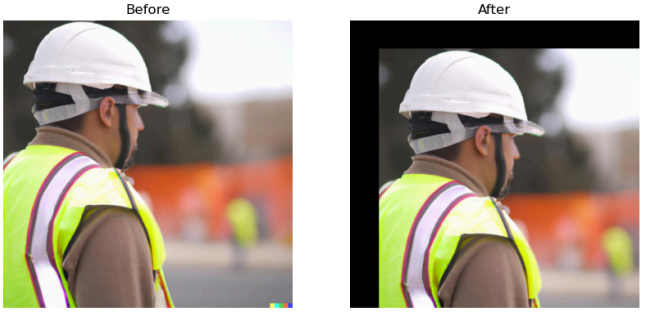

Fig 2. Shifted Image

3. Rotating an image.

Fig 3. Rotated Image

4. Flipping an image (vertically or horizontally).

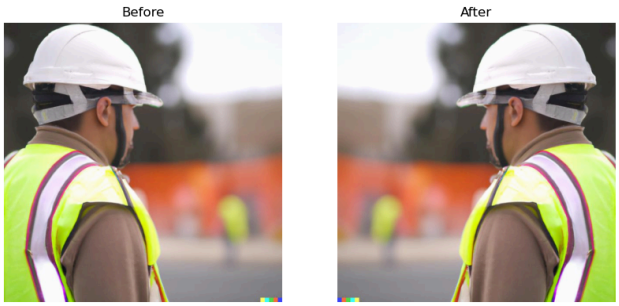

Fig 4. Flipped Image

5. Adjusting the image saturation (increasing or decreasing).

Fig 5. Saturated Image

Each technique was applied to every image in the training set with random parameters, resulting in 1,616 original images expanded to 9,696 images (1,616 images x 5 variations + the original 1,616 images). The test set remained unchanged, with 404 images to ensure consistent evaluation.

After augmentation, DataLoaders were created for efficient data processing. The training DataLoader used a batch size of 32 and had 303 batches, with shuffle set to true to ensure diverse mini-batches during training. The test DataLoader also used a batch size of 32, resulting in 13 batches, with shuffle set to false to maintain consistent evaluation order.

E. Model Architecture

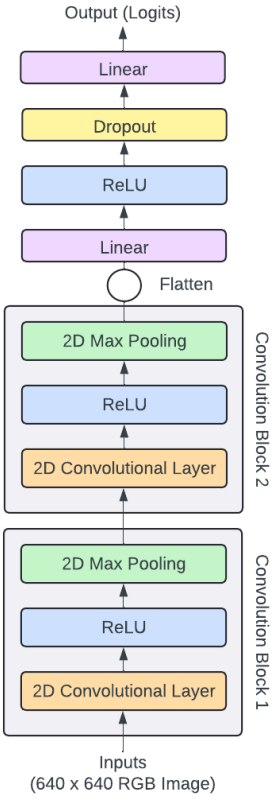

Fig 6. SafetyCNN Model Architecture

TABLE I. SafetyCNN Parameters

Layers	Output Shape	Parameters
Input	3 x 640 x 640	-
2D Convolution 1	12 x 636 x 636	912
ReLU	12 x 636 x 636	-
2D Max Pool	12 x 318 x 318	-
2D Convolution 2	24 x 314 x 314	7224
ReLU	24 x 314 x 314	-
2D Max Pool	24 x 157 x 157	-
Flatten	591,576	-
Linear	64	37,860,928
ReLU	64	-
Dropout	64	-
Linear	1	
Total Parameters	-	37,869,129

The proposed model, SafetyCNN, is a custom convolutional neural network (CNN) designed specifically for workplace safety compliance classification. The network processes input images of size 640x640x3 (height, width, and RGB channels) through two convolutional layers, followed by fully connected layers for final classification. Below is a detailed breakdown of the architecture.

The first convolutional layer applies 12 filters of size 5x5, with a stride of 1 and no padding, to the input image. The output dimensions of this layer are calculated using the formula [7]:

$$Convolution\ Output\ Dimension = [\frac{I-F+2P}{S}] + 1 \times D$$

Where:

- I = 640 (input size),
- F = 5 (kernel size),
- S = 1 (stride),
- P = 0 (padding),
- D = 12 (number of filters/output channels).

This results in an output of 636x636x12. A max pooling operation with a kernel size of 2x2 and stride 2 reduces this output to 318x318x12, halving both spatial dimensions.

The second convolutional layer applies 24 filters of size 5x5, again with stride 1 and no padding, producing an output of 314x314x24. Following max pooling, the output dimensions are reduced to 157x157x24.

The feature maps are then flattened, resulting in 591,576 features (157×157×24), which are passed to a fully connected layer with 64 neurons. A dropout layer with a probability of 0.5 is applied to reduce overfitting. The final layer outputs a single logit for binary classification (safe or unsafe), which is passed through the BCEWithLogitsLoss for training. The model is then trained using the Adam optimizer with a learning rate of 0.001 and weight decay of 1e-4.

The SafetyCNN architecture draws inspiration from a paper [5] that developed lightweight convolutional networks for safety compliance tasks. Their focus on simplicity, efficiency, and generalization through techniques like dropout informed the two-layer convolutional design of SafetyCNN. This approach ensures robust performance while maintaining computational efficiency for real-world safety monitoring applications.

F. Model Training

TABLE II. SafetyCNN Hyperparameters for Training

Hyperparameters	Value
Batch Size	32
Epochs	10
Criterion	BCEWithLogitsLoss
Optimizer	Adam
Learning Rate	0.001
Weight Decay	0.0001

The SafetyCNN model was trained for 10 epochs using the Adam optimizer with a learning rate of 0.001 and weight decay of 1e-4. The BCEWithLogitsLoss criterion was used to compute the loss, while the model parameters were updated through backpropagation. Each training epoch involved a forward pass to compute predictions, loss calculation, and a backward pass to update weights. A batch size of 32 was used, and the training process was monitored using a running average of the loss for performance tracking. The model was trained on a GPU that was available to optimize computation.

G. Model Evaluation with Test-Time Augmentation

Test-Time Augmentation (TTA) enhances the robustness of model predictions by applying various augmentations to the test data and averaging the outputs. This technique mitigates the impact of variability in test images, improving model reliability under diverse real-world conditions [8].

In this study, TTA was implemented with four augmentations:

1. No transformation (original image).
2. Horizontal flipping.
3. Random rotation by up to 30 degrees.
4. Random resizing and cropping (scale: 80%–100%).

Each test image was subjected to these augmentations, and predictions were generated for every augmented version. The SafetyCNN model outputs for each augmentation were averaged to obtain the final prediction probabilities. A threshold of 0.5 was used for binary classification (safe/unsafe).

Evaluation metrics, including accuracy, precision, recall, F1-score, and loss, were calculated on the train and test set using these TTA-enhanced predictions. The process ensured a more reliable performance assessment by accounting for image variations, which is critical for safety compliance scenarios.

H. Convolutional Layers Analysis

To understand the inner workings of the SafetyCNN, the convolutional layers were analyzed using the following methodologies:

1. **Visualizing Convolutional Layers:** Input images were passed through the convolutional and pooling layers sequentially. The feature maps generated at each convolutional layer were visualized to observe how the network extracts spatial features. Each filter's output was plotted as an individual map using matplotlib.
2. **Inspecting Kernel Weights:** The learned weights (kernels) of the convolutional layers were extracted and printed.
3. **Heatmap Visualization for Convolutional Layers:** Layer 1 heatmaps of the learned weights were created for each filter and input channel. Numerical values were annotated on the heatmaps to observe the weight magnitude and

distribution. For Layer 2, due to the higher number of filters, heatmaps were limited to the first two output channels. Each channel's heatmaps were plotted for all input channels to maintain clarity while focusing on prominent patterns.

These analyses offered a detailed view of the feature extraction process and weight distributions, providing valuable insights into the network's learning dynamics.

I. API Deployment

The SafetyCNN model was deployed using FastAPI to enable real-time image classification via a RESTful API. The API allows users to upload an image, which is processed and classified as "Safe" or "Unsafe" using the model with Test-Time Augmentation (TTA).

The deployment was containerized using a Dockerfile and hosted on HuggingFace Spaces, ensuring scalability and accessibility. The application includes endpoints for image classification, leveraging FastAPI's simplicity for seamless interaction. This setup enables practical deployment of the model for workplace safety compliance monitoring.

III. EXPERIMENTAL RESULTS

A. Train Loss Over Time

The SafetyCNN model was trained for 10 epochs, with a consistent reduction in training loss observed over time, indicating effective learning. The average loss decreased from **0.7185** in the first epoch to **0.1091** in the tenth epoch. This steady decline demonstrates the model's ability to minimize error and adapt to the training data effectively. The results highlight the successful optimization of the network parameters over the training period.

B. Train Data Evaluation

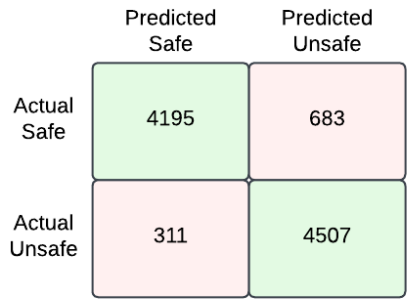

Fig 7. Train Data Confusion Matrix

The SafetyCNN model was then evaluated on the training dataset using Test-Time Augmentation (TTA), achieving strong performance metrics:

TABLE III. Train Data Performance Metrics

Train Loss	0.6226
Accuracy	89.75%
Precision	86.84%
Recall	93.55%
F1 Score	90.07%

The high recall indicates the model's effectiveness in minimizing false negatives, which is crucial for identifying unsafe images accurately. The confusion matrix shows that the model correctly classified 4,195 safe images and 4,507 unsafe images, with relatively fewer misclassifications. These results highlight the model's reliability and strong alignment with the training data.

C. Test Data Evaluation

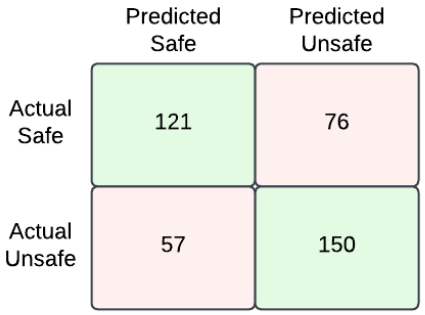

Fig 8. Test Data Confusion Matrix

The SafetyCNN model was also evaluated on the test dataset using Test-Time Augmentation (TTA). The performance metrics on unseen data were as follows:

TABLE IV. Test Data Performance Metrics

Test Loss	0.6756
Accuracy	67.08%
Precision	66.37%
Recall	72.46%
F1 Score	69.28%

The test results reveal a moderate drop in performance compared to the training dataset. While the recall remains relatively high, indicating the model's ability to correctly identify unsafe images, the precision and accuracy decreased. This suggests that the model occasionally misclassified safe images as unsafe, possibly due to differences between the training and test datasets.

In comparison with the training evaluation, the model's performance on the test data highlights a potential overfitting issue, where the model performs better on the training set than on unseen data. This discrepancy emphasizes the need for further regularization or data augmentation strategies to improve generalization. Despite this, the relatively high recall

remains an important strength, as correctly identifying unsafe images is critical in workplace safety applications.

D. Sample Classification with TTA

The model classifies images using Test-Time Augmentation (TTA) to improve prediction robustness. For each test image, multiple augmented versions are generated using transformations such as horizontal flipping, rotation, and random cropping. The model predicts probabilities for each augmented version, which are then averaged to obtain the final probability. A threshold of 0.5 determines the binary classification: **"Safe"** or **"Unsafe."**

Fig 9. Sample Image Prediction

In the example shown in Figure 9, the model correctly classified the image as unsafe, with a sigmoid probability of 0.9461. Despite both individuals wearing helmets, the absence of reflective vests rendered the image non-compliant with worksite safety standards, demonstrating the model's effectiveness in identifying violations.

E. Convolutional Layer 1 Analysis

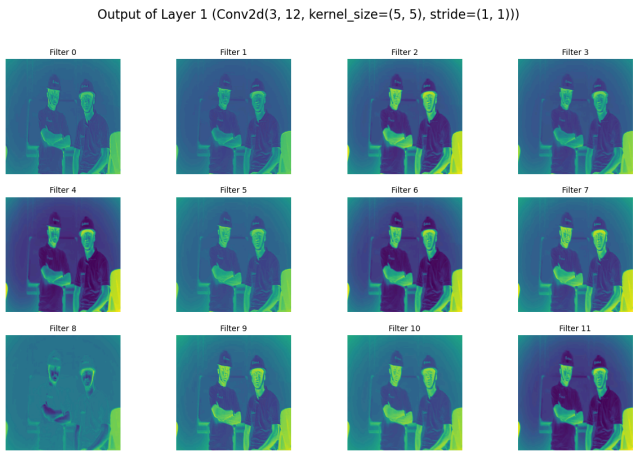

Fig 10. Convolutions of the 1st Convolutional Layer

The first convolutional layer of the SafetyCNN model demonstrates significant diversity in its learned weights, as can be seen in the filter differences in Figure 10, reflecting its ability to capture various spatial patterns from the input images. The layer consists of 12 filters, each spanning three input channels (RGB), with the following key observations:

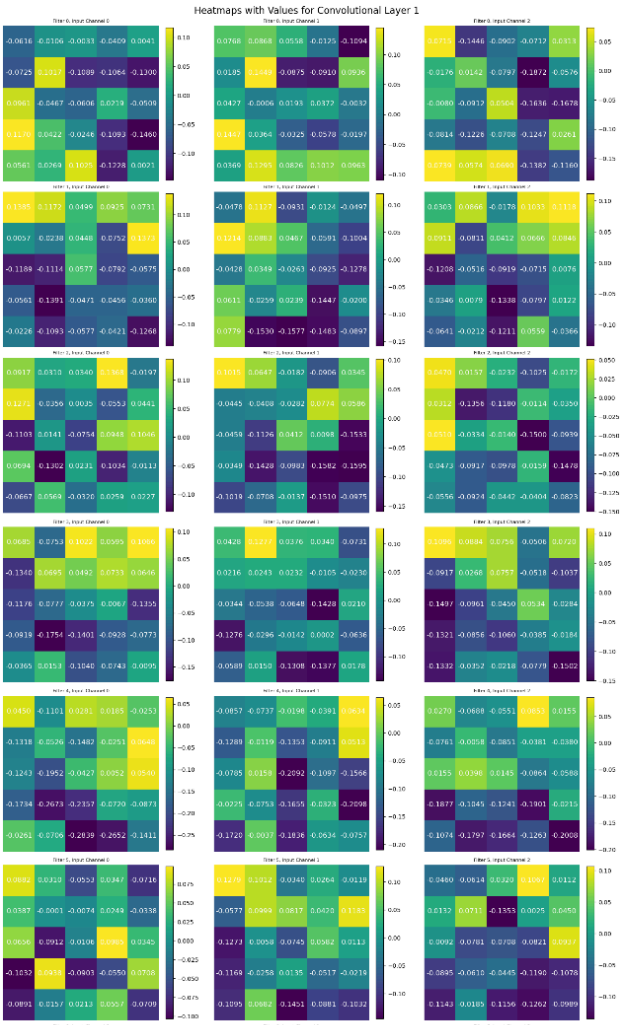

Fig 11. Filters 0 to 5 of the 1st Convolutional Layer

1. **Diverse Spatial Pattern Learning:** Filters exhibit a wide range of weights, indicating the model's ability to detect distinct features such as edges, textures, and gradients. For instance, Filter 0, Input Channel 0, has weights ranging from -0.0616 to 0.1169, capturing both bright and dark patterns in the RGB channel. This capability aligns with findings in edge detection research, where CNNs have been shown to achieve human-level performance by learning rich and abstract edge representations [9].
2. **Filter Specialization:** Certain filters show a high degree of sensitivity to specific features. For example, Filter 1, Input Channel 0 focuses on pronounced patterns with weights ranging between -0.1391 and 0.1372, while others, such as Filter 2, Input Channel 1, demonstrate subtler variations (-0.1533 to 0.0774).
3. **Symmetry in Weight Distribution:** Weights in filters like Filter 4, Input Channel 0 are symmetrically distributed around zero ([-0.1101,

-0.0526, -0.0251, 0.0648]), which is indicative of a focus on uniform patterns such as textures or gradients. Research on symmetrical filters in CNNs indicates that imposing symmetry constraints can reduce the number of free parameters while maintaining performance, suggesting that symmetry plays a role in efficient pattern detection [10].

4. **Channel Variance:** Weights across the second and third channels exhibit distinct patterns compared to the first, underscoring the model's ability to differentiate between RGB features.

Fig 12. Filters 6 to 11 of the 1st Convolutional Layer

5. **Extreme Weight Values:** Certain filters, such as Filter 10 and Input Channel 2, display highly negative values (-0.1962, -0.1772), suggesting over-sensitivity to specific patterns. While this highlights the model's adaptability, it may lead to instability or overfitting.

F. Convolutional Layer 2 Analysis

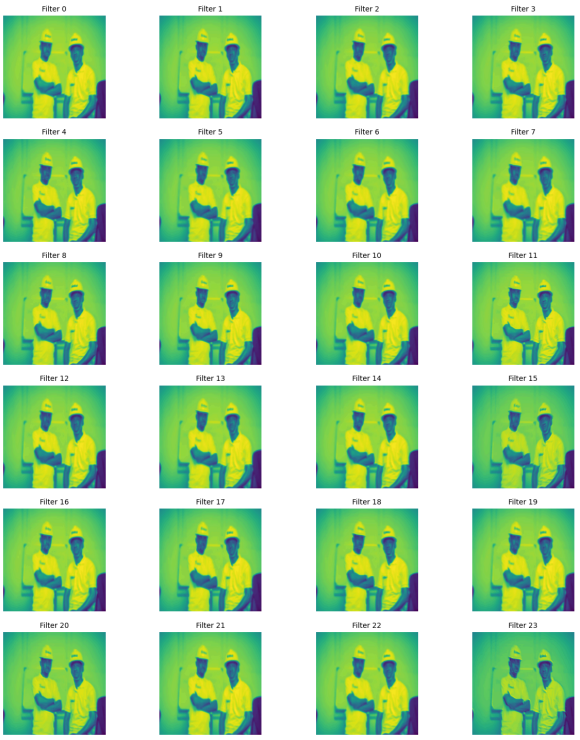

Fig 13. Convolutions of the 2nd Convolutional Layer

The second convolutional layer of the SafetyCNN model, containing 24 filters with 12 input channels each, was analyzed to understand its role in feature extraction and pattern detection. For clarity, the analysis focused on the first two filters, which highlight the following:

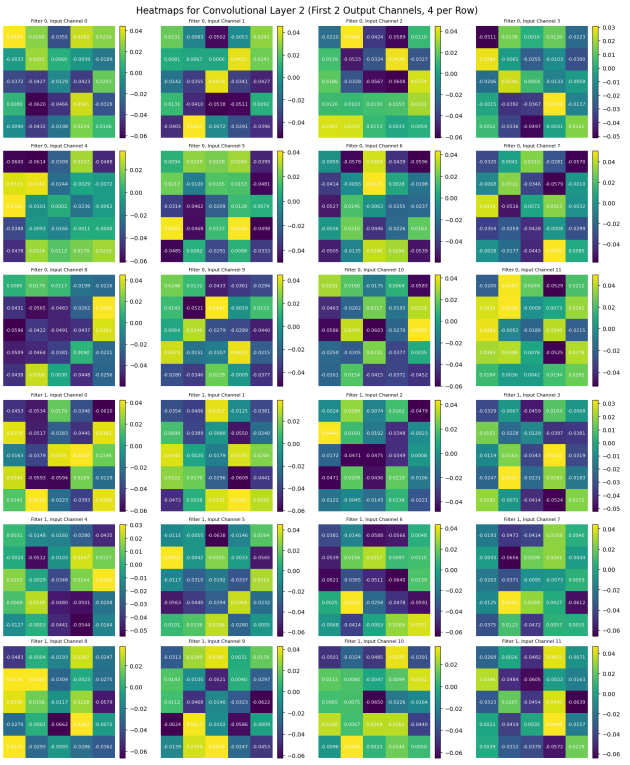

Fig 14. Filters 0 and 1 of the 2nd Convolutional Layer

1. **Intensity-Based Filters:** Filter 0, Input Channel 8, and Filter 1, Input Channel 6 exhibit predominantly negative weights, indicating an emphasis on detecting dark or low-intensity regions. Such focus may assist in identifying shadowed or less illuminated patterns in construction site images [11].
2. **Filter Collaboration in Edge Detection:** Filter 0, Input Channel 3 (-0.0511, 0.0138, 0.0161) and Filter 1, Input Channel 3 (-0.0329, 0.0102, 0.0183) both display symmetry in weights, focusing on uniform patterns and edge structures. These filters likely work in tandem to detect edges at various scales or orientations.

IV. CONCLUSION

This study demonstrates that a simple convolutional neural network (SafetyCNN) can effectively classify construction site images for safety compliance, emphasizing the critical need to minimize false negatives. By prioritizing recall, the model ensures that unsafe scenarios are correctly identified, reducing the risk of overlooking hazardous situations. The incorporation of data augmentation techniques and Test-Time Augmentation (TTA) significantly enhanced the model's generalization capabilities, leading to a high recall rate of 93.55% on the training set and 72.46% on the test set.

The analysis of convolutional layers revealed that the model effectively learns diverse spatial patterns and features relevant to detecting personal protective equipment. The visualization of feature maps and kernel weights provided insights into the network's decision-making process, confirming its ability to focus on critical visual cues associated with safety compliance.

Despite the moderate drop in performance on the test data, the model maintained a strong emphasis on recall, aligning with the primary objective of minimizing false negatives in safety-critical applications. The results highlight the potential of leveraging simple CNN architectures with effective augmentation strategies to develop cost-effective and scalable solutions for workplace safety monitoring.

V. FUTURE WORKS

Future research can build upon this foundation in several ways:

1. **Hyperparameter Tuning and Regularization:** Exploring advanced hyperparameter optimization techniques and regularization methods, such as cross-validation or adding batch normalization.
2. **Integration of State-of-the-Art Techniques:** Comparing and enhancing the current model with state-of-the-art architectures, like deeper CNNs or transformer-based models, may yield better performance while assessing computational trade-offs.
3. **Real-Time Detection Implementation:** Extending the model for real-time video analysis would enhance practical applicability.

VI. REFERENCES

[1] S. P. Breloff, E. Garza, A. Brogan, J. Bunting, D. Trout, M. Pena, and G. S. Earnest, "Struck-by injuries in the construction sector: Common hazards, barriers, and opportunities to keep workers safe," Centers for Disease Control and Prevention, Apr. 4, 2023. [Online]. Available: https://blogs.cdc.gov/niosh-science-blog/2023/04/04/2023-struck-by-stand-down/

[2] Bureau of Labor Statistics, "Construction deaths due to falls, slips, and trips increased 5.9 percent in 2021," U.S. Department of Labor, May 1, 2023. [Online]. Available: https://www.bls.gov/opub/ted/2023/construction-deaths-due-to-falls-slips-and-trips-increased-5-9-percent-in-2021.htm

[3] M. Bonyani and M. Soleymani, "Towards improving workers' safety and progress monitoring of construction sites through construction site understanding," arXiv.org, Oct. 27, 2022. [Online]. Available: https://arxiv.org/abs/2210.15760

[4] M. S. Islam, S. Shaqib, S. S. Ramit, S. A. Khushbu, A. Sattar, and S. R. H. Noori, "A deep learning approach to detect complete safety equipment for construction workers based on YOLOV7," arXiv.org, Jun. 11, 2024. [Online]. Available: https://arxiv.org/abs/2406.07707

[5] M. a. R. Alif, "Enhancing construction site safety: a lightweight convolutional network for effective helmet detection," arXiv.org, Sep. 19, 2024. [Online]. Available: https://arxiv.org/abs/2409.12669

[6] *"Worksite Safety Monitoring Dataset," Kaggle, Nov. 16, 2024. [Online]. Available: https://www.kaggle.com/datasets/lbquctrung/worksite-safety-monitoring-dataset/*

[7] V. Kohir, "Calculating Output dimensions in a CNN for Convolution and Pooling Layers with KERAS," Medium, Dec. 14, 2021. [Online]. Available: https://kvirajdatt.medium.com/calculating-output-dimensions-in-a-cnn-for-convolution-and-pooling-layers-with-keras-682960c73870

[8] M. Kimura, "Understanding Test-Time augmentation," arXiv.org, Feb. 10, 2024. [Online]. Available: https://arxiv.org/abs/2402.06892

[9] Z. Su, W. Liu, Z. Yu, D. Hu, Q. Liao, Q. Tian, M. Pietikäinen, and L. Liu, "Pixel difference networks for efficient edge detection," 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 510-519. doi: 10.1109/ICCV48922.2021.00507.

[10] G. Dzhezyan and H. Cecotti, "SYMNET: Symmetrical filters in convolutional neural networks," arXiv preprint, arXiv:1906.04252, Jun. 10, 2019. [Online]. Available: https://arxiv.org/abs/1906.04252.

[11] N. Maitlo, N. Noonari, S. A. Ghanghro, S. Duraisamy, and F. Ahmed, "Color recognition in challenging lighting environments: CNN approach," in 2022 IEEE 7th International Conference for Convergence in Technology (I2CT), 2024. doi: 10.1109/I2CT61223.2024.10543537.