Knowledge Distillation-based modeling for Camera Obstruction Detection using a custom Dataset

Emmanuel George P
IIIT Kottayam

Aileen Emelda V
IIITDM Kancheepuram

Dr. Masilamani V
IIITDM Kancheepuram

IEEE CICT 2025
19–21 December 2025

Outline

1. Motivation
2. Background
3. Problem Statement
4. Objectives
5. System Architecture/Mathematical Formulation
6. Key Results
7. Discussion
8. Conclusion

Motivation

- CCTV surveillance cameras, though extensively deployed for security in diverse environments, are highly susceptible to both intentional and accidental obstructions, which can render surveillance systems ineffective.
- The traditional reliance on human operators to constantly monitor live feeds for such blockages is not only time-consuming but also impractical, especially within large camera networks.
- Since CCTV surveillance data is sensitive information sending data to the cloud for processing is not ideal, thus we need a model that can run on edge devices with a decent accuracy.

Background

- Vikas et al. proposed ResNet-based models with GAN-augmented data to detect mud and water camera blockages, achieving high accuracy but lacking scalability to diverse CCTV obstructions and requiring heavy computation.
- Hsing Hung et al. used knowledge distillation with Gramian matrix and KL-divergence, improving lightweight student accuracy on CIFAR-100 while significantly reducing model size and computational complexity.

Background

- Hyeonseok Hong et al. introduced feature distribution–based knowledge distillation using KL-divergence demonstrating notable accuracy improvements across multiple architectures on the CIFAR-100 benchmark.
- Gyeongdo Ham et al. proposed cosine similarity–based distillation with adaptive temperature scaling, enabling effective teacher–student alignment and achieving improved accuracy over baseline models on CIFAR-100

Problem Statement

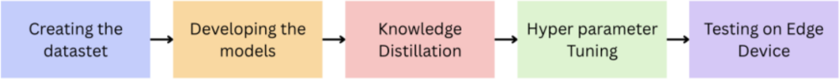
Obstructions in CCTV camera surveillance can render systems ineffective, and manual detection becomes impractical when monitoring a large number of cameras. This work proposes a camera obstruction detection system designed to run on low-resource devices, preserving data privacy by eliminating the need to transmit data to the cloud.

Objectives

- Create an image dataset containing two classes: normal and blocked.
- Develop lightweight models using knowledge distillation based on feature and KL-divergence losses.
- Implement hyperparameter tuning to find optimal hyperparameters and compare results.
- Test model inference time on edge devices

System Architecture/Mathematical Formulation

Overall system architecture dataset creation, developing models, knowledge distillation, hyperparameter tuning and finally testing on edge devices



System Architecture/Mathematical Formulation

The dataset used here was created by the following methods

- Frames were extracted from the CCTV surveillance video dataset
- Some images were created manually by blocking the camera and clicking the pictures
- Blocking by the hand was also stimulated by taking images from the 11K hands dataset
- Blocked images were also created using the diffusion models with appropriate prompts

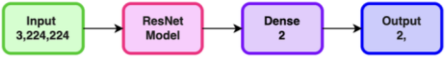
System Architecture/Mathematical Formulation

Some samples of the images in the dataset

(a) Normal Sample 1 (b) Normal Sample 2 (c) Blocked Sample 1 (d) Blocked Sample 2

System Architecture/Mathematical Formulation

Two models, teacher and student models were designed with the following model architectures.



(a) Teacher Model Architecture

(b) Student Model Architecture

System Architecture/Mathematical Formulation

The KL divergence based knowledge distillation was done on these proposed models. The KL divergence loss was found out and added to the cross entropy loss function.

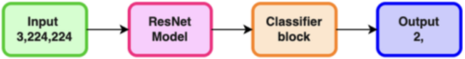
$$\mathrm{Softmax}_T(z) = \mathrm{softmax}\left(\frac{z}{T}\right)$$

$$\mathcal{L}_{\mathrm{KD}} = T^2 \cdot D_{\mathrm{KL}}\left(\mathrm{Softmax}_T(z_t) \,\|\, \mathrm{Softmax}_T(z_s)\right)$$

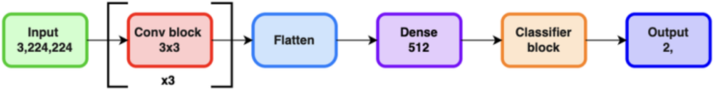
$$\mathcal{L}_{\mathrm{total}} = \alpha\,\mathcal{L}_{\mathrm{CE}} + (1-\alpha)\,\mathcal{L}_{\mathrm{KD}}$$

System Architecture/Mathematical Formulation

In Order to implement feature based knowledge distillation the models were altered to give identical intermediate feature outputs.



(a) Teacher Model Architecture



(b) Student Model Architecture

System Architecture/Mathematical Formulation

Three types of loss functions, MSE, Cosine Similarity and Combined which is combination of both them, were used to implement feature based knowledge distillation

$$\mathcal{L}_{\text{feature-MSE}} = \|f_t - f_s\|_2^2$$

$$\mathcal{L}_{\text{feature-cos}} = 1 - \frac{\langle f_t, f_s \rangle}{\|f_t\|_2 \, \|f_s\|_2}$$

$$\mathcal{L}_{\text{feature-combined}} = \beta \, \mathcal{L}_{\text{feature-MSE}} + (1 - \beta) \, \mathcal{L}_{\text{feature-cos}}$$

$$\mathcal{L}_{\text{total}} = \alpha \, \mathcal{L}_{\text{CE}} + (1 - \alpha) \, \mathcal{L}_{\text{feature}}$$

System Architecture/Mathematical Formulation

Comparison of number of parameters in the student and teacher model

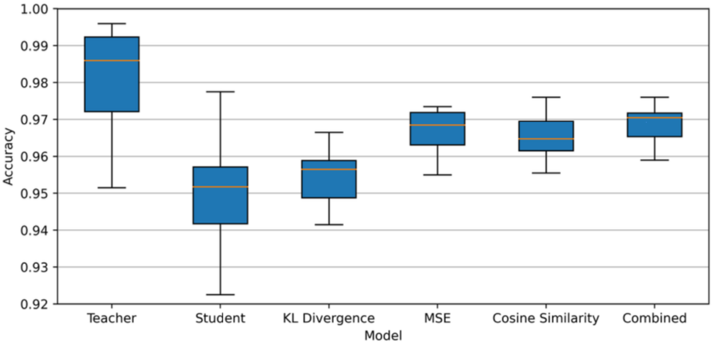
TABLE III
MODEL PARAMETER COUNTS FOR DIFFERENT CONFIGURATIONS.

Model	Parameters (Millions)
Teacher - KL	11.1
Student - KL	2.1
Teacher - Feature	12.3
Student - Teacher	1.2

System Architecture/Mathematical Formulation

- The KL divergence model has 2 hyperparameters which are T and alpha while the feature based models have one that is alpha except for the combined model which has an additional hyperparameter beta.
- Further we tested on multiple different combinations of these hyperparameters using cross validation and found out the best performing combination.
- Then for testing on edge devices we converted the model to onnx format and ran it on the raspberry pi.

Key Results



Box plot showing accuracy values for all 10 runs of cross validation

Key Results

TABLE IV
BEST HYPER PARAMETER VALUES

Model Name	α	β	T
KL-Divergence KD Model	0.35	-	3
MSE KD Model	0.75	-	-
Cosine Similarity KD Model	0.45	-	-
Combined KD Model	0.35	0.3	-

TABLE V
AVERAGE ACCURACY OF THE MODELS

Model	Accuracy (%)
Teacher Model	98.02
Student Model	94.92
KL-divergence KD Model	95.51
MSE KD Model	96.71
Cosine Similarity KD Model	96.52
Combined KD Model	**96.88**

Tables showing best hyper parameter values and the average accuracies of all the models

Key Results

TABLE VI

AVERAGE INFERENCE TIME (IN MILLISECONDS) FOR STUDENT AND TEACHER MODELS UNDER DIFFERENT SYSTEM CONFIGURATIONS

System Configuration		Inference Times (in ms)		Speedup
CPU	RAM	Student	Teacher	
Raspberry Pi 2B	1GB	142.3	1738.0	12.21
intel i3 2012	4GB	9.6	79.6	8.29
M3 Pro Apple	18GB	1.3	10.7	8.23
P100 GPU	16GB	0.7	2.2	3.14

Table showing inference time comparisons of the teacher and student model

Discussion

- The MSE loss accounts for the magnitude of the feature vector and the cosine similarity loss accounts for the direction of the feature vector in the feature space thus combined loss function gives higher accuracy.
- The student model achieved 142.3 ms inference time, with 12.21× speedup and 10.25× model size reduction.

Conclusion and Future

- A new dataset of 14,000 CCTV images was introduced, classified into Blocked and Normal categories.
- Future work will focus on expanding the dataset, improving robustness under diverse conditions.
- Combined MSE and cosine similarity loss model achieved the best accuracy of 96.88% after optimization with the small model thus suitable for deployment on low resource systems.
- Other model compression techniques like quantization can be implemented on the models.

Thank

Thank You!

Questions and discussion are welcome.

emmanuel22bcs104@iiitkottayam.ac.in