Journal of Computer Science

Original Research Paper

Face Log Creation from Low-Light CCTV Videos

Somasundaram Sony Priya and Rajasekharan Indra Minu

Department of Computing Technologies, SRM Institute of Science and Technology, Kattankulathur, India

Article history
Received: 08-08-2024
Revised: 11-10-2024
Accepted: 24-10-2024

Corresponding Author:
Rajasekharan Indra Minu
Department of Computing
Technologies, SRM Institute of
Science and Technology,
Kattankulathu, India
Email: minur@srmist.edu.in

Abstract: In today’s rapidly evolving technological landscape, surveillance systems have become critical for security and operational management. Extracting accurate facial data from low-light CCTV footage remains a significant challenge due to limited visibility. This research presents a comprehensive methodology to address the complexities of face detection, recognition and timestamp extraction in low-light environments. Our approach focuses on creating detailed face logs with in-time and out-time information for each identified individual. The methodology leverages the Enhanced Deep Curve Estimation (EDCE) technique to improve visibility, followed by the Dual Shot Face Detector (DSFD) for precise face detection in enhanced video frames. FaceNet is employed for robust face recognition, while a combination of the Kalman filter and tesseract OCR enables accurate face tracking and timestamp extraction. All extracted data, including facial details and timestamps, are systematically logged into an Excel file for further analysis. The integration of these techniques offers significant advancements in overcoming the challenges of face identification in low-light conditions, presenting a promising solution for enhanced surveillance systems.

Keywords: Video Enhancement, Dual Shot Face Detector, FaceNet, Kalman Filter, Tesseract OCR

Introduction

The increasing reliance on surveillance systems for security and operational management has elevated the need for effective automated video analysis to accurately identify individuals from CCTV recordings. This task becomes particularly challenging in low-light environments, where facial features are often obscured. Traditional face detection methods, such as the Viola-Jones algorithm (Viola and Jones, 2001), rely on handcrafted features and classifiers but are limited in dynamic and complex conditions, especially in poor lighting, occlusions and variations in facial posture. Similarly, the Histogram of Oriented Gradients (HOG) (Dalal and Triggs, 2005) method has been widely used for feature extraction, but it struggles with pose and lighting variations in real-world low-light scenarios.

In response to these limitations, deep learning-based approaches have emerged as more robust alternatives, capable of learning hierarchical features directly from data. Convolutional Neural Networks (CNNs) form the backbone of modern face detection and recognition systems (Zheng and Gupta, 2022). Models such as the Single Shot Detector (SSD) (Liu *et al.*, 2016) and Multi-Task Cascaded Convolutional Networks (MTCNN) (Zhang *et al.*, 2016) have improved face detection accuracy across varying scales. However, even advanced models like RetinaFace (Deng *et al.*, 2020) and the Dual Shot Face Detector (DSFD) (Li *et al.*, 2019) face difficulties in low-light environments, where noise and reduced visibility degrade their performance.

To address the challenges of low-light face detection, image enhancement techniques such as Zero-reference Deep Curve Estimation (ZeroDCE) (Guo *et al.*, 2020) and EnlightenGAN (Jiang *et al.*, 2021) have been proposed. While these methods enhance image visibility, they are computationally expensive and can introduce noise, leading to false positives during face detection. Additionally, these methods are not fully integrated into face detection and recognition pipelines designed specifically for surveillance purposes.

Recent studies have explored illumination-invariant models and robust feature extraction techniques to improve performance under varying lighting conditions. Despite these advances, models like HLA-Face (Wang *et al.*, 2021) and RetinaFace (Deng *et al.*, 2020) still encounter limitations in extremely low-light environments.

In terms of face recognition, models such as FaceNet (Schroff *et al.*, 2015) efficiently map faces into high-

Science Publications

dimensional feature vectors for identification. However, their effectiveness is often constrained by the quality of face detection in low-light environments oupled with the challenge of tracking detected faces across frames using methods like the Kalman filter (Kalman, 1960), robust face detection and recognition in low-light scenarios require a more integrated approach that combines these techniques effectively.

This study introduces a comprehensive methodology designed to improve face detection, recognition and timestamp extraction from low-light CCTV footage. Our approach integrates Enhanced Deep Curve Estimation (EDCE) for image enhancement, the Dual Shot Face Detector (DSFD) (Li *et al*., 2019) for accurate face detection and FaceNet (Schroff *et al*., 2015) for robust face recognition. Additionally, the Kalman filter is employed to track faces across video frames, while Tesseract OCR extracts in-time and out-time information for each detected face, resulting in accurate event logging for surveillance footage.

Related Work

Face identification is a complex task that has been addressed through various techniques, from traditional computer vision algorithms to more advanced deep learning approaches. One of the earliest successes in this area was the Viola-Jones algorithm (Viola and Jones, 2001), which used Haar-like features in conjunction with the AdaBoost learning algorithm to detect faces in images. While effective in constrained settings, its reliance on a limited set of features often resulted in challenges when faced with diverse facial variations, leading to the detection of non-face regions as faces. This limitation becomes particularly problematic in scenarios involving occlusions or variations in lighting.

To address these limitations, Dalal and Triggs (2005) introduced the Histogram of Oriented Gradients (HOG) method, which focuses on gradient orientation and magnitude to capture edge and contour information. Although HOG improved detection performance over earlier methods, it continued to struggle in more complex conditions, such as significant variations in facial expression, lighting and pose. Other traditional methods, such as Scale-Invariant Feature Transform (SIFT) and Local Binary Patterns (LBP) (Ahonen *et al*., 2006), also advanced face detection but faced similar challenges with lighting changes, occlusions and differentiating between individuals with similar appearances.

The rise of deep learning revolutionized face recognition by enabling models to automatically learn complex facial features directly from raw image data, making them adaptable to diverse and challenging conditions. A key breakthrough in this field was Facebook’s DeepFace (Taigman *et al*., 2014), which used convolutional and fully connected layers to achieve impressive accuracy in face recognition, even under varying lighting and pose conditions. With a 97.35% accuracy on the Labeled Faces in the Wild (LFW) dataset, DeepFace set a new benchmark in face recognition. However, it required large amounts of labeled data and significant computational power, making it less practical for low-resource environments.

Building on these advancements, Multi-Task Cascaded Convolutional Networks (MTCNN) (Zhang *et al*., 2016) were introduced to enhance both face detection and alignment. MTCNN employs three cascaded CNNs (P-Net, R-Net and O-Net) to generate candidate bounding boxes, refine those boxes and detect facial landmarks, improving the accuracy of face alignment. Despite its effectiveness, MTCNN is resource-intensive, making it difficult to deploy in environments with limited computational power. Similarly, the Single Shot Detector (SSD) (Liu *et al*., 2016) was designed to enable real-time face detection using VGG16 as its backbone network. While SSD achieved fast detection speeds, it struggled with extreme poses and occlusions, sometimes producing false positives.

The PyramidBox model (Tang *et al*., 2018) introduced a novel approach by utilizing pyramid-based feature extraction to detect faces at multiple scales and resolutions, even in low-resolution and occluded conditions. While PyramidBox achieved strong performance, its high computational demand limited its practicality in low-resource environments. Several researchers have since focused on improving face detection accuracy by addressing the challenge of detecting faces of varying sizes. For instance, the Deep Pyramid Single Shot Face Detector (DPSSD) (Ranjan *et al*., 2019) significantly enhanced the ability to detect large-scale facial variations but required days of training, making it unsuitable for rapid deployment. Suma *et al*. (2021) proposed a method that combines Haar-cascade classifiers with LBP and CNN to detect faces under various lighting and pose conditions. While this approach improved accuracy, it was relatively slow, requiring 4.67 seconds per image, which is too slow for real-time applications.

Recent models, such as FRNetFuse (Huang and Chen, 2022), have aimed to improve face recognition under challenging conditions, particularly low-light environments. FRNetFuse first detects and crops faces using MTCNN before applying Dynamic Histogram Equalization (DHE) to enhance illumination. The images are then processed using a Feature Restoration Network (FRNet) to generate embeddings for face recognition. While promising, this approach remains computationally intensive due to the denoising and enhancement stages, making it less suitable for real-time or large-scale deployments.

Face recognition has evolved significantly from traditional handcrafted feature extraction methods like Eigenfaces and Fisherfaces (Belhumeur *et al*., 1997) to deep learning-based approaches. Eigenfaces, which employed Principal Component Analysis (PCA), offered a computationally efficient way to reduce the dimensionality of face images, but it struggled to handle lighting, pose variations and facial expressions. Fisherfaces, utilizing Linear Discriminant Analysis (LDA), improved robustness to lighting changes by enhancing class separability, but still encountered difficulties with occlusions and significant pose changes.

To address these challenges, texture-based methods like Local Binary Patterns (LBP) (Ahonen *et al*., 2006) were developed. LBP improves robustness to lighting changes by comparing pixel intensities, creating a histogram to represent facial texture. However, LBP and other traditional methods such as SIFT and HOG struggled to generalize in real-world conditions, particularly under extreme pose variations and occlusions. These limitations paved the way for deep learning-based models that could learn more complex and abstract facial features directly from data.

DeepFace (Taigman *et al*., 2014) marked a major leap forward by leveraging Convolutional Neural Networks (CNNs) to automatically learn hierarchical features, significantly improving accuracy under difficult conditions. However, the high computational costs and the large amounts of training data required limited its scalability. FaceNet (Schroff *et al*., 2015) further revolutionized the field by introducing an embedding-based model that employed a triplet loss function to minimize the distance between embeddings of the same person while maximizing the distance between different individuals. This embedding-based approach allowed for highly efficient face verification, recognition and clustering, making FaceNet a widely adopted model in various real-world applications due to its accuracy and efficiency.

Another important model, VGGFace (Parkhi *et al*., 2015), used a deeper CNN architecture (VGG16) to learn facial representations from millions of labeled images. Although VGGFace performed well in face recognition tasks, its training requirements and computational demands remained high, similar to DeepFace. More recent models, such as ArcFace (Deng *et al*., 2019) and CosFace (Wang *et al*., 2018), have introduced new approaches for improving the discriminative power of face embeddings. ArcFace introduced an additive angular margin loss to improve intra-class compactness and inter-class separability, while CosFace used a large margin cosine loss to enhance the decision boundary between classes.

Despite the progress in face detection, tracking and recognition, there is limited research that seamlessly integrates timestamp extraction from surveillance videos, especially in low-light conditions. Current methods often treat face tracking and timestamp extraction as separate processes, lacking an integrated approach to associate faces with specific timestamps in challenging environments.

Materials and Methods

The methodology employed in this study addresses the challenges of face detection in low-light video environments and the extraction of in-time and out-time for each detected face. The experimental process follows several critical steps. First, a dataset of low-light videos is collected and prepared for analysis. The Enhanced Deep Curve Estimation (EDCE) technique (Sony Priya and Minu, 2023) is applied to improve visibility in the video frames. Following this, the Dual Shot Face Detector (DSFD) (Li *et al*., 2019) is used to accurately detect faces within the enhanced footage. Detected faces are then recognized using FaceNet (Schroff *et al*., 2015). Optical Character Recognition (OCR) with Tesseract (Smith, 2007) is employed to extract timestamp information for each detected face. Finally, all detected face details and their corresponding timestamps are recorded and saved in an Excel file. Figure (1) illustrates the complete workflow of the proposed methodology. The subsequent sections will provide further detail on each of these steps.

Video Enhancement

Video enhancement is crucial for improving face detection performance in low-light environments. In such low-light conditions, the quality of video footage is often compromised, resulting in reduced visibility and clarity of facial features. Utilizing video enhancement techniques significantly enhances the caliber of the input data, which improves the visibility of facial features and, in turn, enhances the effectiveness of face detection algorithms or models. In this study, we employ the EDCE (Enhanced Deep Curve Estimation) model, (Sony Priya and Minu, 2023) to improve the quality of low-light CCTV recordings.

The EDCE model consists of seven convolutional layers, each containing 32 filters with dimensions of 3×3 and a stride of 1. The ReLU activation function is utilized in all layers except for the final one, which employs the Tanh activation function. The model processes an input image to produce higher-order curves for image enhancement. Equation (1) is used to derive the enhancement curve, capturing detailed variations in the input image.

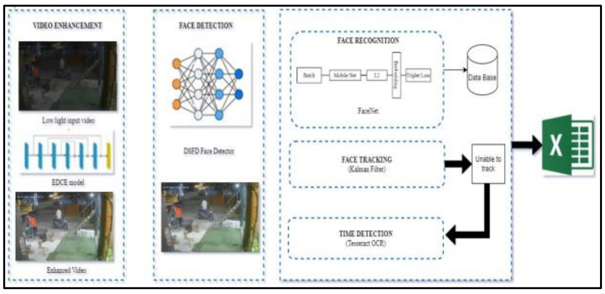


Fig. 1: Overall architecture

$$EC\,(p_{input}(j);\lambda) = p_{input}\,(j) + \lambda\, p_{input}\,(j)^2\,(1\text{-}p_{input}\,(j))^2 \tag{1}$$

In Eq. (1), $p_{input}\,(j)$ represents the pixel in the input picture, 1-$p_{input}\,(j)$ represents its counterpart. The parameter λ, learned during training, regulates the enhancement applied to the low-light input image. Higher values of λ result in a stronger enhancement effect, whereas lower values yield a more modest augmentation.

To utilize the enhancement equation in video processing, the initial step is the conversion of the video into separate frames. Subsequently, the frames undergo processing using EDCE model to achieve enhancement. The enhancement curve is iteratively applied to each frame until the desired level of enhancement is achieved. Before commencing the iterative procedure, a preliminary threshold value of 0.3 is established. The enhancement procedure will terminate outputs the final improved image when the difference between the enhanced frames of the current and previous iterations falls below a predetermined threshold. If the discrepancy exceeds the specified threshold, the procedure proceeds with another iteration. Figure (2) depicts the whole framework for video improvement. The processed frames quality is improved using different loss functions, such as Spatial Consistency Loss (L_{scl}), Color Constancy Loss (L_{ccl}), Exposure Control Loss (L_{ecl}) and Total Variation Loss (L_{tvl}). The overall loss function is thereafter defined as:

$$L_{total} = L_{scl} + w_{ccl}\,L_{ccl} + L_{ecl} + w_{tvl}\,L_{tvl} \tag{2}$$

Here, w_{ccl} and w_{tvl} are weight factors.

Face Detection and Recognition

For each video frame, the algorithm first employs a face detection algorithm, DSFD (Li *et al*., 2019) to identify faces present in the frame. For face recognition transfer learning approach is used. It attempts to find a match for the detected face within the database. If a match is found, the person's name associated with the face is assigned. Otherwise, it is marked as unknown.

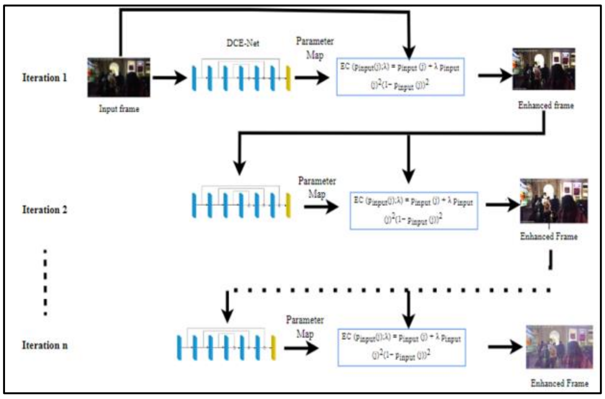

Iteration 1
DCE-Net
Parameter Map
EC (Pinput(j);λ) = Pinput (j) + λ Pinput (j)²(1− Pinput (j))²
Input frame
Enhanced frame
Iteration 2
Parameter Map
EC (Pinput(j);λ) = Pinput (j) + λ Pinput (j)²(1− Pinput (j))²
Enhanced Frame
Iteration n
Parameter Map
EC (Pinput(j);λ) = Pinput (j) + λ Pinput (j)²(1− Pinput (j))²
Enhanced Frame

Fig. 2: Video Enhancement using EDCE

Additionally, the time in the CCTV video is recorded as the in-time for the newly detected face and added to the in-times list. If the face is already being tracked, as indicated by its presence in the current face detected list, the algorithm updates its position using a face-tracking algorithm Kalman Filter (Kalman, 1960) This helps to accurately track the face's movement across frames. In cases where a face cannot be tracked i.e., it is not available in the consecutive frames, the algorithm marks the time as out-time which is added to the out-times list. After processing all video frames, the algorithm creates an Excel file to store the detected face, person's name, in-time and out-time.

DSFD Face Detection Model

DSFD (Li *et al*., 2019) tailored for face detection. It excels in handling faces across diverse perspectives and challenging conditions, making it particularly suitable for face detection in surveillance video scenarios. The model operates within a dual-shot detection framework, incorporating two essential components: The Feature Enhance Module (FEM) and Improved Anchor Matching (IAM). FEM strategically leverages information from different levels, thereby enhancing feature discriminability and robustness. IAM employs advanced techniques, such as partitioning strategies for anchors and data augmentation based on anchors. These enhancements refine the matching of anchors with ground truth faces, leading to superior regressor initialization and, consequently, more precise and accurate face detection. The DSFD framework, employs the same backbone network as the SSD network. A notable distinction lies in the transformation of six feature maps at different depths into six "enhanced" feature maps, achieved through Feature Enhance Module. The two shots use different loss functions to capture small and large faces. Loss function for the second shot (L_{ssl}) is defined as:

$$L_{ssl}(\widehat{p}_k, \widehat{m}_k, \hat{t}_k, g_k, a_k) = \frac{1}{N_{conf}}\sum_k L_{conf}\,(\widehat{p}_k, \widehat{m}_k) + \beta\,\frac{1}{N_{loc}}\sum_k \widehat{m}_k\, L_{loc}(\hat{t}_k\,, g_k, a_k) \tag{3}$$

where, $\widehat{p}$ predicted confidence score for the anchor, $\widehat{m}$: Binary indicator (0 or 1) denoting whether the anchor is positive, $\hat{t}$: Predicted bounding box parameters for the anchor, g: Ground-truth bounding box for the object, a: Anchor. β: Weight parameter that is utilized to equalize the effects of the classification and localization components. L_{conf} is the SoftMax loss used to measure how well the model predicts the class probabilities for each anchor. L_{loc} represents a smooth $\mathcal{L}_1$ loss function that quantifies the discrepancy between the predicted bounding box and the actual ground truth. This loss function incentivizes the model to precisely estimate the spatial position and dimensions of the object (face) within the anchor. Loss function for the first shot (L_{fsl}) for small anchors (sa) is defined in Eq. (4). Finally, L_{ssl} and L_{fsl} are combined to find the total loss which is known as Progressive Anchor Loss (L_{pal}) which is depicted in Eq. (5):

$$L_{fsl}(\widehat{p}_k, \widehat{m_k}, \widehat{t_k}, g_k, sa_k) = \frac{1}{N_{conf}} \sum_k L_{conf}(\widehat{p}_k, \widehat{m_k}) + \beta \frac{1}{N_{loc}} \sum_k \widehat{m}_k L_{loc}(\widehat{t_k}, g_k, sa_k) \tag{4}$$

$$L_{pal} = L_{fsl}(sa) + \alpha L_{ssl}(a) \tag{5}$$

FaceRecognition Using FaceNet

FaceNet (Sony Priya and Minu, 2023) developed by Google Researchers, revolutionizes face recognition by compressing a person's face into a 128-dimensional vector through the feature embedding function $g(x)$. This function maps an input image x into a feature space $\mathbb{R}^d$ with the goal of reducing the squared distance between embeddings of identical identities while increasing the separation between embeddings of distinct identities. This property facilitates accurate face recognition across varying imaging conditions. This is mathematically formulated as:

$$\| f(x^i_{anchor}) - f(x^i_{positive}) \|^2 + \gamma < \| x^i_{anchor}) - f(x^i_{negative}) \|^2 \tag{6}$$

Here:

$f\ (x^i_{anchor}\ )$ represents the feature vectors of the anchor image.

$f\ (x^i_{positive})$ represents the feature vectors of all other positive images of the same person.

$f\ (x^i_{negative})$ represents the feature vectors of any negative images of any other person.

γ is a margin enforced between positive and negative pairs.

This formulation captures the essence of the desired relationship, ensuring that in the feature space, the anchor image is positioned nearer to positive images of the same individual compared to negative images of different individuals, maintaining a minimum margin of γ. During training, triplets (*anchor*, *positive*, *negative*) are sampled from the dataset. Then the embedding function (f) is used to map the examples to the feature space. Then the distance between the embeddings is found using the Triplet loss function. Which is depicted as:

$$L_{triplet} = max\,(\|\ f\ (x^i_{anchor}) - f\ (x^i_{positive})\ \|^2 - \| x^i_{anchor}) - \mathrm{f}\,(x^i_{negative}) \|^2 + \gamma, 0) \tag{7}$$

In summary, *Triplet* loss encourages the model to learn embeddings that capture the inherent structure of the data, making it suitable for tasks where relative similarity is crucial.

FaceTracking Using Kalman Filter

Tracking faces in complicated settings can provide a challenge owing to camera noise and fluctuating lighting conditions. The Kalman Filter (KF) (Kalman, 1960) is utilized to forecast the face's location in consecutive frames. KF is a corrective predictor method. The core principle of the Kalman filter revolves around utilizing former state information to predict the subsequent state. To track the face, the position is characterized by its coordinates (x, y). The state vector, denoted as 'x,' encapsulates vital information encompassing both the facial position and its associated velocity. More formally, it takes the form $x = [x_{position}, y_{position}, x_{velocity}, y_{velocity}]$. The prediction phase involves projecting the anticipated state of the system for the upcoming time step, leveraging the information from the previous state and accounting for inherent system dynamics. The following equations are key components of this predictive process:

State Prediction: The predicted state, denoted as 'x_p', is obtained by multiplying the state transition matrix 'F' with the current state 'x'.

$$x_p = F^* x \tag{8}$$

Here, 'x_p' serves as an approximation of the forthcoming state, while 'F' represents the state transition matrix that encapsulates the evolution of the state over time.

Covariance prediction. The state covariance matrix 'P' undergoes an update based on the prediction. This update is facilitated by considering the influence of 'F' and the process noise covariance matrix 'Q':

$$P = F^*\ \mathrm{P}^*\ F^T \tag{9}$$

The matrix 'P' now embodies the state covariance, while 'Q' reflects the covariance associated with the process noise. This noise matrix quantifies uncertainties linked with the inherent dynamics of the system.

Update step: Upon formulating predictions, the Kalman filter proceeds to fuse these projections with newly acquired measurements, refining the state estimate. This update phase involves the following equations.

Measurement residual: The residual, denoted as 'y, ' signifies the discrepancy between the observed measurement 'z' (e.g., facial position) and the measurement projection based on the predicted state 'x_p':

$$y = z - H^* x_p \tag{10}$$

Here, 'z' represents the observed state and 'H' denotes the measurement matrix mapping the projected state to the measurement space.

Residual covariance: The covariance matrix 'S' characterizing the measurement residual is determined by Eq. (11):

$$S = H^{*} P^{*} H^{T} + R \tag{11}$$

The matrix '*S*' encapsulates the covariance associated with the measurement residual, while '*R*' captures the uncertainties intrinsic to the measurement process.

Kalman gain: The Kalman gain matrix '*K*' serves as a crucial factor for combining the predictions with measurements. It is derived from the Eq. (12):

$$K = P^{*} H^{T*} S^{-1} \tag{12}$$

State update: The final state update takes into account the Kalman gain '*K*' and the measurement residual '*y*,' incorporating the corrections from the measurement to refine the estimated state '*x*':

$$x = x_p + K^{*} y \tag{13}$$

Covariance update: The covariance matrix '*P*' is refined further by considering the Kalman gain and the measurement matrix. This process ensures that the covariance matrix effectively reflects the updated state estimate:

$$P = (1 - K^{*} H)^{*} P \tag{14}$$

The prediction and update steps iteratively unfold as new measurements of the facial position are obtained. The Kalman filter adeptly amalgamates these projections and measurements, rigorously accommodating the associated uncertainties to generate an optimal and refined state estimate.

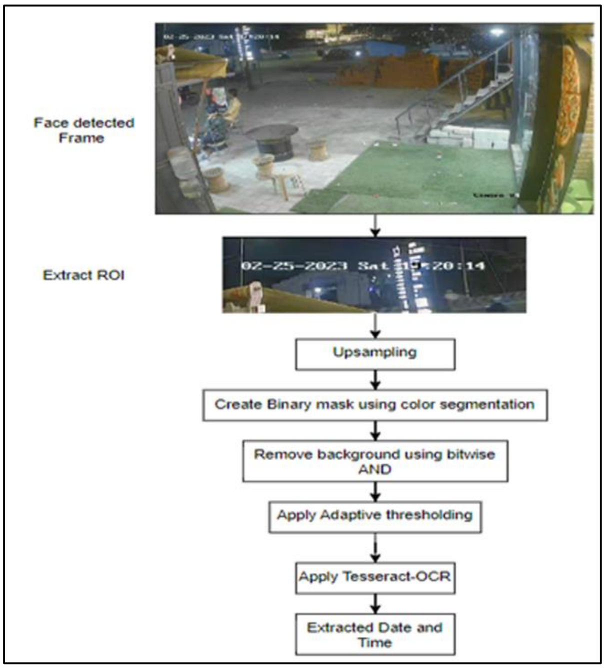

Face detected Frame
Extract ROI
02-25-2023 Sat 15:20:14
Upsampling
Create Binary mask using color segmentation
Remove background using bitwise AND
Apply Adaptive thresholding
Apply Tesseract-OCR
Extracted Date and Time

Fig. 3: Time extraction from CCTV video frames

Time Extraction

To extract time from an image, the process begins by enhancing the image quality to optimize the effectiveness of the Optical Character Recognition (OCR) process using Tesseract. Tesseract OCR (Smith, 2007) is a widely utilized open-source OCR engine known for its versatility and support for multiple languages and platforms. The extraction of date and time is carried out through a series of image processing steps. First, the Region of Interest (ROI) containing the timestamp is cropped and upsampled to improve its resolution. The image is then converted to a binary representation, creating a binary mask where the timestamp is highlighted in white and the background is masked in black. This mask is applied to the original ROI using a bitwise AND operation, isolating the timestamp from the background. Finally, adaptive thresholding is applied to enhance contrast and further separate the timestamp text from the background, allowing accurate extraction of the time details. Figure (3) illustrates these steps.

Results and Discussion

All experiments were conducted in a Google Collab environment with GPU acceleration enabled. We implemented our model using TensorFlow, Kera's and various open-source libraries. In this study, we collected a dataset of low-quality CCTV videos from public places, consisting of 47 videos. Each video is 5 min long and recorded at 30fps. An example of the input frames is shown in Fig. (4).

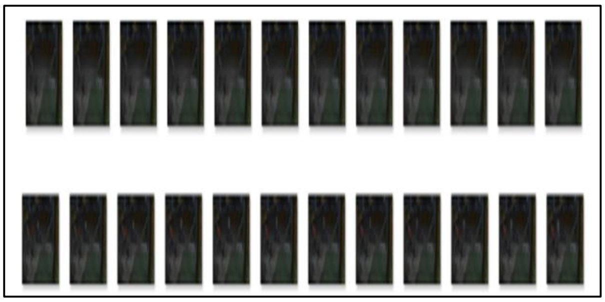


Fig. 4: Input video frames

Fig. 5: Enhanced video frames by EDCE

Figure (5) displays the enhanced video frames produced by the EDCE model. As illustrated, the EDCE model significantly improves the visual clarity of low-light and low-quality CCTV video footage. This enhancement is particularly crucial in surveillance videos where lighting conditions are suboptimal, which can negatively affect the accuracy of subsequent face detection and recognition tasks. By improving contrast, brightness and sharpness, the enhanced frames provide more distinct facial features, which greatly aids the face detection models in identifying facial regions more accurately.

Figure (6) presents the training and validation loss curves during the training phase of the EDCE model. The loss curves plot the model's training performance across epochs, showing both the training loss and the validation loss. The EDCE model was trained for 100 epochs to enhance the video frames. As seen in the figure, the training loss decreases initially, indicating that the model is learning to improve video frame quality over time. The validation loss follows a similar pattern, albeit with some fluctuations. These fluctuations may be due to the variability in the quality of the video frames across different scenes in the CCTV footage, which introduces challenges for the model. Despite these fluctuations, both training and validation losses show a decreasing trend overall, demonstrating that the model is learning effectively and generalizing reasonably well to unseen data. By applying the EDCE model for video enhancement, we were able to improve the quality of input frames used for face detection and recognition, leading to better performance in those subsequent tasks.

After enhancing the video frames using the EDCE model, we applied face detection on the enhanced frames using various pre-trained face detection models, including DSFD (Li *et al*., 2019), MTCNN (Zhang *et al*., 2016), RetinaFace (Deng *et al*., 2020), SSD (Liu *et al*., 2016), YOLOv3 (Chun *et al*., 2020) and HaarCascade (Viola and Jones, 2001), to find which model is suitable for our dataset. Figure (7) presents the detection results, showing the bounding boxes generated by each model on the improved frames. The use of the EDCE model significantly improves face recognition performance by making facial features more discernible, which enhances the reliability of pre-trained models in detecting faces, especially in low-resolution conditions. The Haar Cascade classifier, while effective in controlled environments, struggles in complex scenes or with variable lighting, leading to missed detections. YOLOv3, a real-time detection model, performs well but can miss smaller or partially occluded faces. The SSD model, known for its speed, occasionally produces oversized bounding boxes and may overlook faces in challenging conditions. MTCNN, which employs a multi-stage convolutional network, excels in face alignment and detection but may miss individuals in low-light or crowded scenes. RetinaFace, leveraging landmark detection, shows strong performance but may face limitations when detecting multiple faces in dynamic settings. Lastly, DSFD, designed for large-scale face detection, demonstrates the ability to accurately detect numerous faces, showing the best results for our dataset.

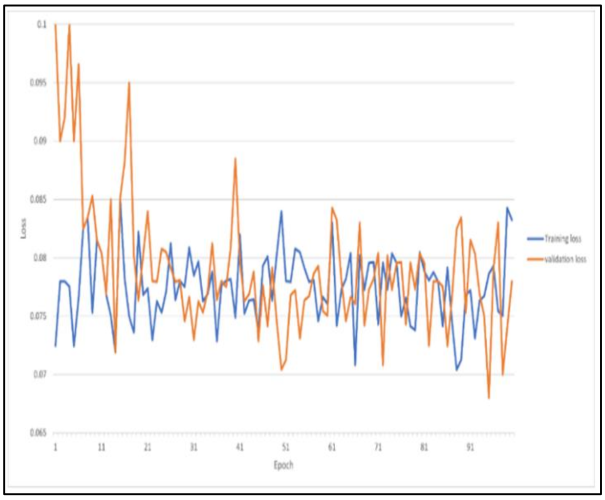


Fig. 6: Training and validation curve for video enhancement by EDCE

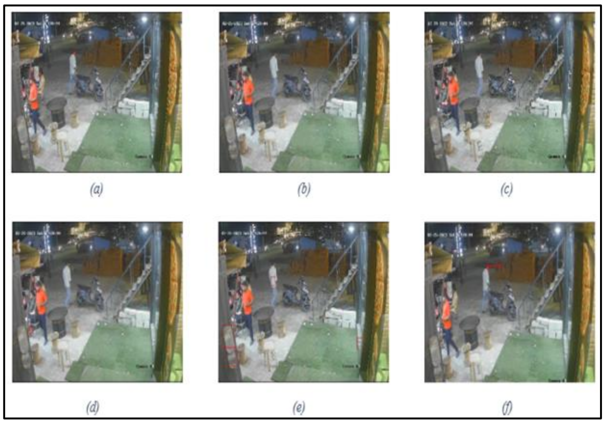


Fig. 7: Face detection results (a) DSFD (b) Retina face (c) MTCNN (d) SSD (e) Haarcascade (f) YOLOV3

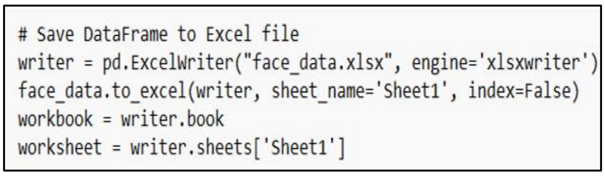
After face detection, embeddings were extracted from each face using deep learning. The FaceNet deep learning model computes a 128-dimensional embedding that quantifies the features of the face. For face recognition, we created a custom database, where each individual is represented by a folder containing four images of their face. This database was used to train the face recognition model, enabling the system to map detected faces to their corresponding identities during the recognition process. The structured database plays a crucial role in ensuring accurate identification of individuals. After training the model on these embeddings, the system is able to recognize faces. In this study, we applied face recognition across different face detection models, with the results shown in Table (1). From the results, it is evident that the combination of DSFD and FaceNet provides the highest accuracy for our dataset, which led us to choose this combination.

Table 1: Performance analysis for face recognition

Model name	Accuracy	Precision	Recall	F1 score
DSFD + FaceNet	94%	0.93	0.92	0.93
MTCNN + Face Net	92%	0.90	0.89	0.89
RetinaFace + FaceNet	94%	0.92	0.91	0.91
SSD + FaceNet	91%	0.89	0.88	0.88
YOLOv3 + FaceNet	89%	0.87	0.85	0.86
HaarCascade + FaceNet	88%	0.86	0.84	0.85

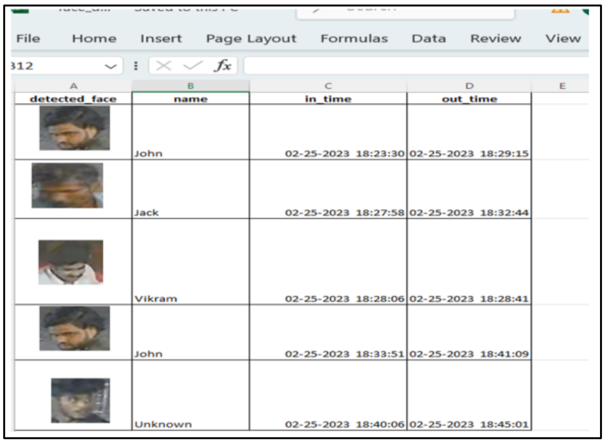
In our model, once a face is detected and enhanced, facial recognition is performed using the FaceNet model, which compares the detected face with a custom database to identify the individual. Concurrently, the Kalman filter tracking algorithm continuously monitors the presence of the recognized face in the video. If the algorithm loses track of the face, the exact moment is recorded as the "out time" for that individual. This process ensures that each recognized face is tracked throughout the video, with accurate entry and exit times logged. All timestamps, along with the recognized faces and corresponding names, are stored in an Excel file for easy reference and further analysis. Figure (8) shows the code used to create the Excel file, while Fig. (9) presents an example of the generated Excel file.

While the model demonstrates strong performance in face detection, recognition and tracking, there are several challenges encountered. First, the accuracy of face detection is compromised by varying facial angles, partial occlusions and low-resolution video frames, which can lead to missed detections or false positives. Sudden movements and changes in lighting conditions further complicate the tracking process, occasionally causing the system to lose track of individuals or misidentify them. This often results in duplicate entries for the same person across different timestamps, which reduces the efficiency of the tracking mechanism. Additionally, the model's reliance on continuous face detection in each frame for tracking purposes creates processing overhead, particularly in crowded or fast-paced environments. This increases the computational complexity and may lead to delayed or inaccurate recognition outputs. Future improvements may focus on enhancing the model's resilience to these challenges through more sophisticated handling of occlusions, dynamic lighting conditions and smoother integration of the tracking and recognition components.



```
# Save DataFrame to Excel file
writer = pd.ExcelWriter("face_data.xlsx", engine='xlsxwriter')
face_data.to_excel(writer, sheet_name='Sheet1', index=False)
workbook = writer.book
worksheet = writer.sheets['Sheet1']
```

Fig. 8: Code snippet to create excel file



detected_face	name	in_time	out_time
	John	02-25-2023 18:23:30	02-25-2023 18:29:15
	Jack	02-25-2023 18:27:58	02-25-2023 18:32:44
	Vikram	02-25-2023 18:28:06	02-25-2023 18:28:41
	John	02-25-2023 18:33:51	02-25-2023 18:41:09
	Unknown	02-25-2023 18:40:06	02-25-2023 18:45:01

Fig. 9: Face log

Conclusion

This study presents an approach for generating face logs from low-light CCTV footage by enhancing visibility and integrating advanced face detection, recognition and timestamp extraction techniques. Leveraging the DSFD face detection algorithm and FaceNet for recognition, our method successfully identifies and tracks individuals in challenging lighting conditions. Additionally, Tesseract OCR is utilized to extract timestamps for each detected face, while a Kalman filter tracks the faces to record their in-time and out-time, with all results systematically logged in an Excel file for analysis. The research makes notable contributions by demonstrating how visibility enhancement can improve the accuracy and reliability of face detection and recognition in low-light surveillance footage. This method is particularly effective in environments where traditional techniques face challenges due to poor lighting, thus advancing the application of facial recognition technologies in complex scenarios.

However, there are some limitations to this method. The system's processing time for video enhancement and face detection presents challenges, particularly for real-time applications. Additionally, the accuracy of Tesseract OCR in extreme low-light conditions and when handling complex backgrounds can be compromised, leading to potential inaccuracies in timestamp extraction. Face tracking can also be disrupted by occlusions, causing erroneous "out-time" entries for individuals.

Future research should aim to optimize the processing pipeline for real-time face detection and recognition. Enhancing the accuracy of OCR under low-light conditions and introducing more advanced face-tracking algorithms, possibly using deep learning, could improve overall performance. Moreover, developing adaptive video enhancement techniques that adjust dynamically to changing lighting conditions would further enhance detection and recognition accuracy. These improvements would broaden the method’s applicability to real-world scenarios, such as public surveillance, traffic monitoring and security operations.

Acknowledgment

We thank all the members who have contributed to this study with us.

Funding Information

The authors received no specific funding for this study.

Author’s Contributions

Somasundaram Sony Priya: Conceptualization, methodology, written-original draft preparation.

Rajasekharan Indra Minu: Conceptualization, formal analysis, written-review and edited.

Both authors read and approved the final manuscript.

Ethics

The authors declare that they have no any ethical issues to report regarding the present study.

References

Ahonen, T., Hadid, A., & Pietikainen, M. (2006). Face Description with Local Binary Patterns: Application to Face Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(12), 2037–2041. https://doi.org/10.1109/tpami.2006.244

Belhumeur, P. N., Hespanha, J. P., & Kriegman, D. J. (1997). Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*(7), 711–720. https://doi.org/10.1109/34.598228

Chun, L. Z., Dian, L., Zhi, J. Y., Jing, W., & Zhang, C. (2020). YOLOv3: Face Detection in Complex Environments. *International Journal of Computational Intelligence Systems*, *13*(1), 1153. https://doi.org/10.2991/ijcis.d.200805.002

Dalal, N., & Triggs, B. (2005). Histograms of Oriented Gradients for Human Detection. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, 886–893. https://doi.org/10.1109/cvpr.2005.177

Deng, J., Guo, J., Ververas, E., Kotsia, I., & Zafeiriou, S. (2020). RetinaFace: Single-Shot Multi-Level Face Localisation in the Wild. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5202–5211. https://doi.org/10.1109/cvpr42600.2020.00525

Deng, J., Guo, J., Xue, N., & Zafeiriou, S. (2019). ArcFace: Additive Angular Margin Loss for Deep Face Recognition. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4685–4694. https://doi.org/10.1109/cvpr.2019.00482

Guo, C., Li, C., Guo, J., Loy, C. C., Hou, J., Kwong, S., & Cong, R. (2020). Zero-Reference Deep Curve Estimation for Low-Light Image Enhancement. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1777–1786. https://doi.org/10.1109/cvpr42600.2020.00185

Huang, Y.-H., & Chen, H. H. (2022). Deep Face Recognition for Dim Images. *Pattern Recognition*, *126*, 108580. https://doi.org/10.1016/j.patcog.2022.108580

Jiang, Y., Gong, X., Liu, D., Cheng, Y., Fang, C., Shen, X., Yang, J., Zhou, P., & Wang, Z. (2021). EnlightenGAN: Deep Light Enhancement Without Paired Supervision. *IEEE Transactions on Image Processing*, *30*, 2340–2349. https://doi.org/10.1109/tip.2021.3051462

Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, *82*(1), 35–45. https://doi.org/10.1115/1.3662552

Li, J., Wang, Y., Wang, C., Tai, Y., Qian, J., Yang, J., Wang, C., Li, J., & Huang, F. (2019). DSFD: Dual Shot Face Detector. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5055–5064. https://doi.org/10.1109/cvpr.2019.00520

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). SSD: Single Shot MultiBox Detector. In B. Leibe, J. Matas, N. Sebe, & M. Welling (Eds.), *Computer Vision – ECCV 2016* (Vol. 9905, pp. 21–37). Springer International Publishing. https://doi.org/10.1007/978-3-319-46448-0_2

Ranjan, R., Bansal, A., Zheng, J., Xu, H., Gleason, J., Lu, B., Nanduri, A., Chen, J.-C., Castillo, C., & Chellappa, R. (2019). A Fast and Accurate System for Face Detection, Identification and Verification. *IEEE Transactions on Biometrics, Behavior and Identity Science*, *1*(2), 82–96. https://doi.org/10.1109/tbiom.2019.2908436

Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2015). Deep Face Recognition. *Procedings of the British Machine Vision Conference 2015*, 1–12.

Schroff, F., Kalenichenko, D., & Philbin, J. (2015). FaceNet: A Unified Embedding for Face Recognition and Clustering. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 815–823. https://doi.org/10.1109/cvpr.2015.7298682

Smith, R. (2007). An Overview of the Tesseract OCR Engine. *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, 629–633. https://doi.org/10.1109/icdar.2007.4376991

Sony Priya, S., & Minu, R. I. (2023). Augmenting Face Detection in Extremely Low-Light CCTV Footage Using the EDCE Enhancement Model. *Traitement Du Signal*, *40*(6), 2741–2750. https://doi.org/10.18280/ts.400634

Suma, K., Sunitha, N. V., Suhasini, N., & Shreekumar, T. (2021). Dense Feature Based Face Recognition from Surveillance Video Using Convolutional Neural Network. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, *12*(5), 1436–1449. https://doi.org/10.17762/turcomat.v12i5.2040

Taigman, Y., Yang, M., Ranzato, M., & Wolf, L. (2014). DeepFace: Closing the Gap to Human-Level Performance in Face Verification. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 1701–1708. https://doi.org/10.1109/cvpr.2014.220

Tang, X., Du, D. K., He, Z., & Liu, J. (2018). PyramidBox: A Context-Assisted Single Shot Face Detector. In V. Ferrari, M. Hebert, C. Sminchisescu, & Y. Weiss (Eds.), *Computer Vision -- ECCV 2018* (Vol. 11213, pp. 812–828). Springer International Publishing. https://doi.org/10.1007/978-3-030-01240-3_49

Viola, P., & Jones, M. (2001). Rapid Object Detection Using a Boosted Cascade of Simple Features. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, 1–1. https://doi.org/10.1109/cvpr.2001.990517

Wang, W., Yang, W., & Liu, J. (2021). Hla-face: Joint high-low adaptation for low light face detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 16195-16204).

Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z., & Liu, W. (2018). CosFace: Large Margin Cosine Loss for Deep Face Recognition. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5265–5274. https://doi.org/10.1109/cvpr.2018.00552

Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. *IEEE Signal Processing Letters*, *23*(10), 1499–1503. https://doi.org/10.1109/lsp.2016.2603342

Zheng, S., & Gupta, G. (2022). Semantic-Guided Zero-Shot Learning for Low-Light Image/Video Enhancement. *2022 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, 581–590. https://doi.org/10.1109/wacvw54805.2022.00064