Occlusion-Resilient Face Detection for Real-World Robustness Using Efficient Machine Learning Techniques

Asmith, Saksham Agnihotri, Dhoop Patel, S.P. Raja*, Kalki Eshwar D

Vellore Institute of Technology, Vellore (India)

unir
LA UNIVERSIDAD
EN INTERNET

ABSTRACT

Face detection in real-world scenarios often faces challenges like occlusions, disorientation, and camouflage, which can reduce the accuracy of traditional methods. This study evaluates seven machine learning models aimed at improving occlusion-resilient face detection. These models include CNNs with data augmentation, YOLO-Face, DETR, transfer learning (e.g., FaceNet), GANs for face reconstruction, an autoencoder with a custom loss function, and a Hybrid CNN-RNN architecture. Each model is tested for its ability to detect partially obscured faces across diverse real-world scenarios using metrics like accuracy, precision, recall, and robustness. Results show that models integrating spatial and temporal features, such as the hybrid CNN-RNN, excel in dynamic environments, while DETR performs well with static images. This evaluation provides insights into the strengths and weaknesses of each model, offering guidance on selecting the best face detection approach based on real-world application needs.

KEYWORDS

Face detection, occlusion-resilient models, Convolutional Neural Networks (CNN), You Only Look Once (YOLO)-Face, Detection Transformer (DETR), Transfer learning (FaceNet), Generative Adversarial Networks (GAN), Autoencoder, Hybrid CNN–RNN, Machine Learning, Real-World applications, Accuracy and Robustness Evaluation, Occlusion Handling

I. Introduction

Face detection is a crucial element in various computer vision applications such as security systems, surveillance operations, and biometric identification. While traditional face detection models have achieved remarkable results under ideal conditions, they often struggle in challenging real-world scenarios involving occlusions, unusual facial orientations, or camouflage.

Occlusion-resilient face detection aims to accurately identify faces that are partially hidden or presented at unconventional angles, ensuring robustness and reliability in real-world face recognition systems where full facial visibility is rare. The challenge lies in the complex variability of human facial features, which can be obscured by masks, glasses, shadows, or camouflage, and further complicated by disoriented faces at extreme angles.

This study evaluates the performance of multiple machine learning models to address these challenges, from enhanced Convolutional Neural Networks (CNNs) to advanced approaches like Detection Transformers (DETR) and Generative Adversarial Networks (GANs). We assess each model's ability to detect faces under occlusion and disorientation, providing insights into their strengths and limitations for both dynamic and static environments. Key evaluation metrics such as accuracy, precision, recall, and Intersection over Union (IoU) are used to gauge the models' robustness. By leveraging a diverse dataset covering various occlusion levels and facial orientations, this analysis offers practical recommendations for selecting effective face detection methods tailored to specific application needs.

* Corresponding authors:
asmithkr1314@gmail.com (Asmith), saksham.agnihotri2003@gmail.com (Saksham Agnihotri), dhoop5903@gmail.com (Dhoop Patel), avemariaraja@gmail.com (S.P. Raja), kalkieshward@gmail.com (Kalki Eshwar D).

II. Related Work

Face detection and recognition have been pivotal in various applications, ranging from security systems to human-computer interaction. Traditional methods often struggled with occlusions, varying poses, and illumination changes. Recent advancements in machine learning and deep learning have led to more robust and accurate models capable of handling these challenges.

Early works by Chou and Chen [1] introduced a real-time multi-face detection system utilizing a Naive Bayes classifier implemented on FPGA hardware. Their approach focused on efficient feature extraction and candidate face detection, achieving high accuracy with low memory consumption. Similarly, Wu et al. [2] proposed a convolutional neural network (CNN) cascade for simultaneous face detection and pose estimation, demonstrating that multi-task learning can enhance feature representation and real-time performance.

Cascaded convolutional networks were further explored by Qi et al. [3], who designed a three-stage deep CNN architecture for face detection. They incorporated separable convolutions and residual structures to improve detection accuracy while maintaining real-time processing capabilities. Data augmentation techniques have also been instrumental in enhancing model generalization. Li et al. [4] demonstrated that data augmentation could significantly improve hyperspectral image classification using deep CNNs, a principle that can be extended to face detection tasks.

Ma et al. [5] addressed the issue of occlusions in pedestrian detection by proposing an image-level automatic data augmentation method. Their approach adjusted augmentation policies based on individual image characteristics, leading to improved detection accuracy. In the context of face detection, Zhao et al. [6] improved the YOLO-v4 algorithm by integrating attention mechanisms such as CBAM (Convolutional Block Attention Module), SENet (Squeeze-and-Excitation Network), and CANet (Channel Attention Network). This enhancement allowed the model to focus on essential features, improving detection accuracy in scenarios where faces are partially obscured.

Anusudha and colleagues [7] combined YOLO-V7 with InsightFace to develop a real-time face recognition system capable of handling occluded and disguised faces effectively. Their system leveraged the strengths of both models to improve recognition accuracy. Transformers have also been introduced into object detection through the Detection Transformer (DETR) framework. Fang et al. [8] enriched one-to-many matching in DETRs using feature augmentation, accelerating training convergence and boosting detection performance without increasing inference complexity.

Addressing occlusions in face detection, Yu et al. [9] proposed TransRPPG, a transformer-based approach for 3D mask face presentation attack detection. Their method enhanced liveness detection by capturing spatio-temporal patterns in facial videos. Transfer learning has proven effective in tasks with limited training data. Vrbančič and Podgorelec [10] introduced adaptive fine-tuning, optimizing which layers of a pre-trained model to fine-tune, improving classification accuracy while reducing training time.

In low-light conditions, face recognition performance can degrade significantly. Fan et al. [11] developed Low-FaceNet, which enhances low-light images to improve face recognition accuracy. Wu and Zhang [12] combined MTCNN (Multi-task Cascaded Convolutional Networks) and an improved FaceNet with a modified loss function, achieving high recognition accuracy suitable for access control systems.

Generative Adversarial Networks (GANs) have been utilized for face reconstruction to handle occlusions. Shahreza and Marcel [13] evaluated the vulnerability of face recognition systems to template inversion attacks via 3D face reconstruction. Malakar et al. [14] focused on improving masked face recognition by generating the occluded lower part of the face using image augmentation and CNNs, enhancing recognition accuracy while preserving facial identity. Luo et al. [15] introduced FA-GAN (Face Augmentation GAN), aiming to create deformation-invariant face representations by augmenting faces with varying attributes.

Autoencoders with custom loss functions have been explored to disentangle facial features. Abdolahnejad and Liu [16] proposed a deep autoencoder with an adaptive resolution reconstruction loss, enabling the model to extract specific facial concepts without labeled data. This approach allows for faithful image reconstruction while separating concepts related to different scales.

Hybrid models combining CNNs and other architectures have shown promise in capturing complex patterns. Kumar and Madhavi [17] implemented a stacked Siamese neural network for content-based image retrieval, demonstrating improved retrieval performance. Zhu et al. [18] developed a continuous human activity recognition system using a hybrid CNN–RNN architecture, effectively capturing spatial-temporal patterns in unconstrained environments. Samadiani et al. [19] proposed a model for happy emotion recognition from unconstrained videos using 3D hybrid deep features, achieving higher accuracy by extracting dynamic spatial-temporal features. There has also been a survey conducted by Kouyumdjieva et. al (2020) [20] detailing the various non-imaging based techniques for counting people.

III. Methodologies

The face detection problem becomes particularly challenging when confronted with occluded, disoriented, or camouflaged faces in real-world scenarios. In this section, we describe seven machine learning models designed to address these challenges. Each model leverages distinct deep learning architectures and techniques to enhance robustness against occlusions, ensuring improved detection accuracy under non-ideal conditions such as partial facial obscuration, varying orientations, and environmental distractions.

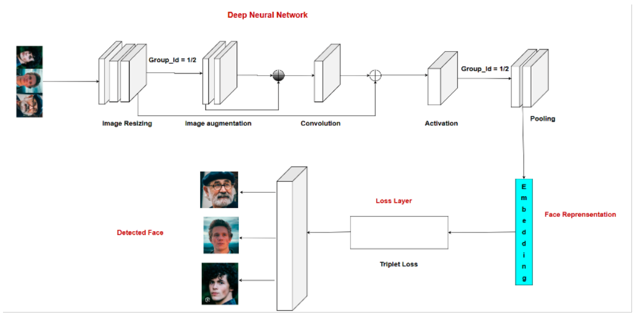

Fig. 1. CNN with Data Augmentation

A. CNN with Data Augmentation

Convolutional Neural Networks (CNNs) are instrumental in extracting hierarchical spatial features essential for face detection. However, their performance can be adversely affected by occlusions and variations in facial orientation. To mitigate this, we incorporate sophisticated data augmentation strategies, drawing inspiration from the works of Li et al. [4] and Ma et al. [5]. By enriching the training dataset with artificially occluded images and applying transformations such as rotations and flips, we enhance the CNN's ability to generalize to occluded faces. This augmentation enables the model to recognize facial patterns despite partial visibility, thereby improving detection accuracy in complex environments. Figure 1 illustrates a CNN with data augmentation.

1. Algorithm 1: CNN with Data Augmentation for Face detection

Input: Image III, augmentation parameters

Output: Detected face bounding boxes BBB and confidence scores CCC

1. **Image Augmentation:** Apply augmentations such as rotation, flip, zoom, and occlusion to create the augmented image.
$$I_{aug} = A(I) \tag{1}$$
2. **Convolutional Layers:** Pass the augmented image through the CNN to generate feature maps.
$$F = Conv(I_{aug}) \tag{2}$$
3. **Pooling Layers:** Down sample the feature maps using max pooling:
$$F_{pooled} = MaxPool(F) \tag{3}$$
4. **Fully Connected Layers:** Flatten and pass the pooled feature maps through fully connected layers to predict bounding boxes and confidence scores:
$$B, C = FullyConnected(F_{pooled}) \tag{4}$$
5. **Output:** Return the bounding boxes B and confidence scores C.

2. Description

In this approach, data augmentation is used to enhance input images by randomly applying transformations such as rotations, flips, zooms, and occlusions. This process generates a more diverse training set for the Convolutional Neural Network (CNN), enabling it to learn face detection under varying

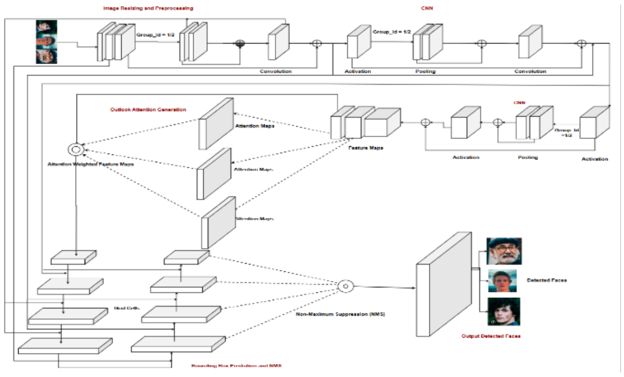

Fig. 2. Technical Flow of YOLO-Face

conditions. The augmented image, denoted as I_{aug}, is fed into multiple convolutional layers where filters extract distinct facial features. These features are progressively abstracted: lower layers capture basic patterns like edges, while deeper layers identify complex facial structures.

To reduce the size of feature maps while preserving essential information, pooling layers—typically max-pooling—are applied. This step decreases computational complexity and helps prevent overfitting by summarizing features in sub-regions. The downsampled feature maps are then flattened and passed through fully connected layers, which predict bounding boxes (BBB) and confidence scores (CCC) for each detected face in the image. By integrating CNN with data augmentation, this method ensures robust performance, making it effective even when faces are occluded or viewed from different angles.

B. YOLO-Face for Face Detection

The You Only Look Once (YOLO) framework is renowned for its real-time object detection capabilities, achieving an optimal balance between speed and accuracy. In this study, we adapt YOLO specifically for face detection under occlusion scenarios by incorporating attention mechanisms inspired by methodologies proposed in Zhao et al. [6] and Anusudha et al. [7]. By integrating modules such as the Convolutional Block Attention Module (CBAM) and Squeeze-and-Excitation Networks (SENet) into the YOLO architecture, the model enhances its ability to prioritize salient facial features while suppressing irrelevant contextual information. This adaptation improves the model's robustness in detecting partially obscured or disguised faces, thereby increasing detection accuracy in practical applications where occlusions are common. Figure 2 illustrates the technical workflow of YOLO-Face.

1. Algorithm 2: The YOLO-Face Detection Algorithm

Input: Image I

Output: Bounding boxes B with confidence scores C

1. **Image Preprocessing:** Scaling and Size
2. Resize the input image I to a x a pixel:

$$I^{`} = resize(I, (a, a)) \tag{5}$$

3. **Normalize the image:**

$$I^{``} = I^{`}/255 \tag{6}$$

4. **Feature Extraction:** Pass the normalized image $I^{``}$ through the CNN layers to extract feature maps F: $F = CNN(I^{``})$
5. **Bounding Box Prediction:** For each grid cell g in feature map F, predict:
6. **Center coordinates (Bx, By):**

$$Bx = \sigma(tx) + cx \tag{7}$$

$$By = \sigma(ty) + cy \tag{8}$$

7. **Width and height (Bw, Bh):**

$$Bw = pw * e^{t}w \tag{9}$$

$$Bh = ph * e^{t}h \tag{10}$$

8. **Convert center coordinates to corner coordinates (x1, y1) (top-left) and (x2, y2) (bottom-right):**

$$x1 = Bx - \frac{Bw}{2} \tag{11}$$

$$y1 = By - \frac{Bh}{2} \tag{12}$$

$$x2 = Bx + \frac{Bw}{2} \tag{13}$$

$$y2 = By + \frac{Bh}{2} \tag{14}$$

9. **Confidence score C:** $C = \sigma(tc)$
10. **Apply Non-Maximum Suppression (NMS):**
11. For each predicted bounding box $Bi = (x1^{i}, y1^{i}, x2^{i}, y2^{i})$ and confidence score Ci apply NMS
12. **Compute IoU between two bounding boxes Bi and Bj:**

$$IoU(Bi, Bj) = \frac{Area(Bi \cap Bj)}{(Area(Bi) + Area(Bj)) - Area(Bi \cap Bj)} \tag{15}$$

13. Retain Bi if IoU (Bi, Bj) < threshold and Ci is the maximum confidence score among overlapping boxes.
14. Return the set of final bounding boxes B and their associated confidence scores C after NMS: $\{Bi = (x1^{i}, y1^{i}, x2^{i}, y2^{i}), Ci\}$ for i = 1, 2, ..., N

2. Description

The initial phase of the YOLO-Face model involves preprocessing the input image to align it with the model's required dimensions. This step entails resizing the image to ensure uniformity across all inputs, maintaining consistency within the dataset. Following resizing, pixel values are normalized to a range between 0 and 1. This normalization standardizes the data, enhancing computational stability and improving model performance by mitigating numerical instabilities during training.

Once preprocessing is complete, the modified image is fed into the YOLO-Face architecture, which utilizes multiple convolutional layers for feature extraction. These layers analyze low-level patterns such as edges, textures, and high-level facial characteristics, forming a hierarchical representation of the input. The extracted features serve as critical inputs for subsequent stages, enabling accurate localization and recognition of faces within the image.

The model then generates multiple candidate bounding boxes to identify potential face regions. Each box is assigned a confidence score reflecting the likelihood of containing a face. This approach allows the model to detect faces at varying positions, scales, and orientations, ensuring robustness under diverse real-world conditions.

To refine these initial predictions, the model employs Non-Maximum Suppression (NMS), a critical post-processing technique. NMS addresses overlapping or redundant bounding boxes by eliminating less confident predictions while retaining those with higher confidence scores and minimal overlap. This process ensures that each detected face is represented by a

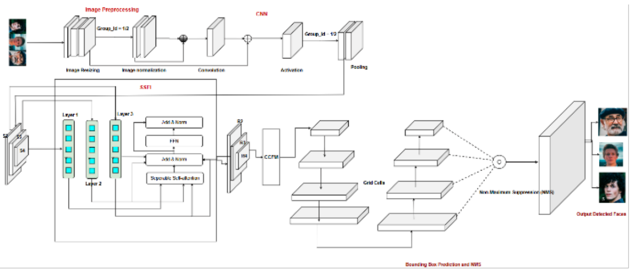

Fig. 3. DETR For Face Detection

single, accurate bounding box, minimizing false positives and improving detection precision.

The final output consists of the remaining bounding boxes after applying NMS. These boxes are scaled back to match the original image dimensions, providing localized regions corresponding to detected faces. Each bounding box includes a confidence score, offering an estimate of the detection's reliability. This culmination yields a set of accurate face annotations, ready for downstream tasks such as identification or tracking in practical applications.

C. DETR (Detection Transformer) for Face Detection

Transformers have revolutionized various domains within machine learning, including object detection tasks. In this study, we implement the Detection Transformer (DETR) framework, building upon the advancements introduced by Fang et al. [8], to perform end-to-end face detection. To enhance the one-to-many matching process intrinsic to DETRs, we incorporate feature augmentation techniques. This modification not only accelerates training convergence but also improves detection accuracy without increasing computational complexity during inference.

The transformer-based architecture of DETR effectively models long-range dependencies and captures contextual relationships among objects, making it particularly effective for detecting faces in cluttered or occluded scenes. By leveraging self-attention mechanisms, the model dynamically weights relevant features, enabling robust localization even when facial regions are partially obscured or distorted. This capability is critical for real-world applications where environmental factors often complicate detection.

Figure 3 illustrates the DETR-based pipeline for face detection, highlighting its integration of feature augmentation and transformer architecture to achieve reliable results under challenging conditions.

1. Algorithm 3: DETR Face Detection

Input: Image III

Output: Detected face bounding boxes BBB with confidence scores CCC

1. **Image Preprocessing:**

$$I^{\grave{}} = resize(I, (a, a)) \tag{16}$$

$$I^{\grave{}\grave{}} = I^{\grave{}}/255 \tag{17}$$

2. **CNN Backbone Feature Extraction:** The normalized image $I^{\grave{}\grave{}}$ is passed through a CNN to generate feature maps F: $F = CNN(I^{\grave{}\grave{}})$
3. **Transformer for Global Context:** The feature maps F are passed into the transformer to capture spatial relationships using multi-head attention: $Z = Transformer(F)$
4. **Object Queries and Bounding Box Prediction:** A set of object queries is used to predict the bounding boxes B and confidence scores C for the faces in the image:

$$Bx = \sigma(tx) + cx \tag{18}$$

$$By = \sigma(ty) + cy \tag{19}$$

$$Bw = pw * e^{t}w \tag{20}$$

$$Bh = ph * e^{t}h \tag{21}$$

$$C = \sigma(tc) \tag{22}$$

5. **Non-Maximum Suppression (NMS):** To remove the overlapping boxes, NMS is applied by computing the intersection over union (IoU):

$$IoU(Bi, Bj) = \frac{Area(Bi \cap Bj}{(Area(Bi) + Area(Bj) - Area(Bi \cap Bj))} \tag{23}$$

6. **Output:** The final set of bounding boxes B and confidence scores C represent the detect faces in the image: $Bi = (x1^{i}, y1^{i}, x2^{i}, y2^{i})$, Ci for i = 1, 2, ..., N

2. Description

The process begins by resizing and normalizing the input image III to a fixed dimension, ensuring consistent preprocessing across all images. This preprocessed image $I'I'I'$ is then passed through a Convolutional Neural Network (CNN) backbone to extract feature maps FFF that capture essential facial features present in the image. While these feature maps are critical for identifying faces, they lack the global contextual information required to understand spatial relationships between different regions of the image.

To address this limitation, the feature maps are fed into a transformer network, which employs its multi-head attention mechanism to model long-range dependencies and contextual relationships across the image. This enables the model to dynamically prioritize relevant facial regions even when faces are disoriented or partially occluded. The transformer's output ZZZ encodes these global dependencies, providing a richer representation that enhances the accuracy of face localization.

Next, the model uses object queries to predict bounding boxes around detected faces. These queries interact with the transformer's output ZZZ, allowing the model to estimate center coordinates, width, height, and confidence scores for each potential face. This approach bypasses traditional region proposal methods by directly predicting bounding boxes based on the comprehensive contextual information provided by the transformer.

Finally, Non-Maximum Suppression (NMS) is applied to refine the detected bounding boxes by eliminating redundant or overlapping predictions. This step ensures that each face is represented by a single, precise bounding box. The final output consists of the predicted bounding boxes BBB and their corresponding confidence scores C which highlight the locations of detected faces in the image.

D. Fine-Tuned Transfer Learning

Transfer learning offers a powerful means to leverage pre-trained models for specialized tasks, especially when data availability is limited. We adopt the FaceNet model, renowned for its expertise in face recognition, and fine-tune it for the specific task of occlusion-resilient face detection. By selectively adjusting layers within the pre-trained network—inspired by the adaptive fine-tuning approach proposed by Vrbančič and Podgorelec [10] and the modified loss function strategy introduced by Wu and Zhang [12] —we optimize the model's

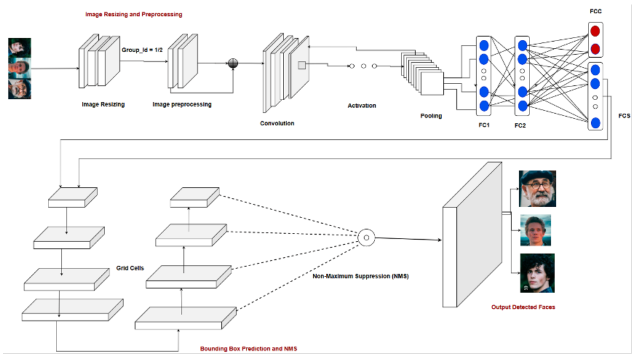

Fig. 4. Fine-Tuned Transfer Learning (e.g. FaceNet)

performance for our application. This fine-tuning process enables the model to retain valuable learned features while adapting to the complexities of detecting occluded faces, thereby enhancing accuracy without requiring extensive retraining from scratch. Figure 4 illustrates the fine-tuned transfer learning framework.

1. Algorithm 4: Transfer Learning Face Detection

Input: Partially occluded/disoriented face image III

Output: Detected face bounding boxes BBB with confidence scores CCC

1. **Load Pre-Trained FaceNet Model**: Load a pre-trained FaceNet model, keeping most layers frozen and preserving learned facial embeddings $FaceNet_{pre-trained}(I)$
2. **Image Preprocessing**: Resize and normalize the input image I for FaceNet's input requirements:

$$I^{'} = resize(I, (a, a)) \tag{24}$$

$$I^{''} = \frac{I^{'}}{255} \tag{25}$$

3. **Feature Extraction:** Pass the preprocessed image through the frozen layers of the pre-trained FaceNet model to extract high-level facial features:

$$F = FaceNet_{frozen}(I^{''}) \tag{26}$$

4. **Fine-Tuning the Model:** Fine-tune the final layers of the model using a smaller face detection dataset, adjusting weights for the task of bounding box prediction:

$$B, C = Fine - tuned - layers(F) \tag{27}$$

5. **Bounding Box Prediction:** The fine-tuned layers output bounding boxes BBB and confidence scores CCC for each detected face:

$$B_x = \sigma(t_x) \tag{28}$$

$$B_y = \sigma(t_y) \tag{29}$$

$$B_w = p_w.e^{tw} \tag{30}$$

$$B_h = p_h.e^{th} \tag{31}$$

$$C = \sigma(t_c) \tag{32}$$

6. **Output Detected Faces:** Return the bounding boxes BBB and confidence scores CCC representing the detected faces.

$$\{Bi = (x1^{i}, y1^{i}, x2^{i}, y2^{i}), Ci\} \tag{33}$$

for i = 1, 2, ..., N

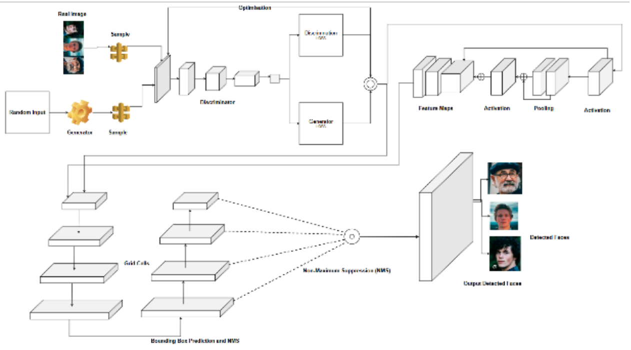

Fig. 5. GAN for Face Reconstruction

2. Description

The fine-tuned transfer learning approach begins by loading a pre-trained FaceNet model, which has already learned high-level facial features. The model is frozen up to a specific layer, retaining its knowledge from large-scale datasets. Only the final layers are adapted to address the task of occlusion-resilient face detection. This adaptation involves training the model on a smaller dataset with labeled face bounding boxes.

Next, input image III is resized and normalized to meet FaceNet's input requirements. The preprocessed image is then passed through the frozen layers of the pre-trained model to extract facial embeddings—high-level representations encoding spatial features of the face. These embeddings are subsequently processed by the fine-tuned layers, which predict bounding boxes B and confidence scores C for each detected face.

Finally, the model outputs the detected face bounding boxes along with their corresponding confidence scores. This transfer learning strategy enables efficient and accurate face detection with minimal training effort, as the pre-trained model provides robust feature representations, while the fine-tuned layers specialize in adapting to the detection task.

E. GAN for Face Reconstruction

Generative Adversarial Networks (GANs) have demonstrated remarkable success in generating realistic images and reconstructing missing or corrupted data. To directly address occlusions in facial images, we implement a GAN-based face reconstruction module, inspired by the methodologies of Shahreza and Marcel [13], Malakar et al. [14], and Luo et al. [15]. This module leverages unoccluded facial images as training data to learn the statistical patterns of facial structures. By iteratively refining its output through adversarial training, the GAN generates plausible reconstructions of occluded regions, effectively restoring the missing parts of the face.

The reconstructed faces provide a more complete representation of facial features, which enhances their utility for subsequent detection and recognition tasks. This process not only improves the quality of input data but also strengthens the system's robustness in handling partial occlusions caused by environmental factors or intentional disguises. The integration of GANs into the pipeline ensures that critical facial details are preserved and amplified, even in challenging scenarios.

Figure 5 illustrates the architecture of the GAN-based face reconstruction module, highlighting its role in restoring occluded regions and improving overall detection accuracy in real-world applications.

1. Algorithm 5: Face Detection Using GAN for Reconstruction and Identification

Input: Partially occluded/disoriented face image III

Output: Detected face bounding boxes BBB with confidence scores CCC

1. **Image Preprocessing:** The input face image I, which may be occluded or disoriented is resized to a fixed dimension a x a:

$$I^{`} = resize(I, (a, a)) \tag{34}$$

2. The pixel values of the resized image $I^{`}I^{`}I^{`}$ are normalized to the range [0, 1]:

$$I^{``} = I^{`}/255 \tag{35}$$

3. **GAN-based Face Reconstruction:** The normalized image $I^{``}$ is passed into the Generator Network GGG, which extracts and reconstructs the missing or occluded parts of the face using an encoder-decoder architecture:

$$F_{reconstructed} = G(I^{``}) \tag{36}$$

4. **Discriminator Evaluation:** The reconstructed face image $F_{reconstructed}$ and a real face image F_{real} are input into the Discriminator network DDD. The Discriminator assigns probabilities D(F_{real}) $\approx$ 1for real images and D($F_{reconstructed}$ $\approx$ 0 for generated images. The Generator aims to minimize the Generator loss:

$$\mathcal{L}_{\mathcal{G}} = -log(D(G(I^{``}))) \tag{37}$$

5. **Optimize the Generator:** Until convergence repeat.

$$Loss : \mathcal{L}_{\ulcorner} = -[log(D(F_{real})) + log(1 - D(F_{reconstructed)})] \tag{38}$$

Update the network

$$\theta_G \leftarrow \theta_G - \eta_G \nabla \theta_G \mathcal{L}_G \tag{39}$$

6. **Face Feature Enhancement:** Through adversial training, the reconstructed face $F_{reconstructed}$ is enhanced by reducing occlusions and correcting disorientations, improving facial feature completeness.
7. **Bounding Box Prediction:** Using $F_{reconstructed}$ a face detection model (e.g., YOLO) predicts bounding boxes B for each grid cell g in the feature map F:

$$Bx = \sigma(tx) + cx \tag{40}$$

$$By = \sigma(ty) + cy \tag{41}$$

The width and height Bw and Bh are predicted as:

$$Bw = pw * e^{t}w \tag{42}$$

$$Bh = ph * e^{t}h \tag{43}$$

The confidence score CCC for each bounding box is calculated:

$$C = \sigma(tc) \tag{44}$$

8. Output Detected Faces: The final set of bounding boxes B with confidence scores C highlights the detected faces: $Bi = (x1^{i}, y1^{i}, x2^{i}, y2^{i}), Ci$ for i = 1, 2, ..., N

2. Description

The process begins with the input face image III which is partially occluded or disoriented. This image is first normalized, scaling its pixel values to lie within the range [0, 1] represented as $I^{''} = I/255$. Through this normalization the input for the model, ensures enhanced stability. The normalized image is then passed through the Generator network GGG, where an encoder-decoder architecture extracts features from the visible portions of the face. The encoder compresses these visible facial features into a latent space, encoding critical details about the face's structure. Meanwhile, the decoder uses this latent space, encoding critical details about the face's structure. Meanwhile, the decoder uses this latent representation to reconstruct the missing or occluded regions of the face. The Generator's output is the reconstructed face image, known as $F_{reconstructed} = G(I^{''})$ which attempts to recreate a complete face that closely resembles a real one. In the context of face detection, this reconstruction enables the model to recover occluded facial features, improving the detection performance on incomplete or disoriented inputs.

Once the Generator produces the reconstructed face image $F_{reconstructed}$, it is passed to the Discriminator network DDD alongside a real face image F_{real}. The Discriminator evaluates both the real and reconstructed images, outputting a probability score that indicates whether each image is real or generated. For the real face image F_{real}, the Discriminator is expected to output a score close to 1, i.e., $D(F_{real}) \approx 1$, signalling that it recognizes the image as a real face. For the reconstructed face $F_{reconstructed}$, the Discriminator should output a score close to 0, i.e., $D(F_{reconstructed}) \approx 0$, indicating that it identifies the image as fake. The goal of the Generator is to fool the Discriminator by minimizing the Generator Loss:

$$L_G = -\log_{f_0}(D(G(I)) \tag{45}$$

This loss drives the Generator to produce reconstructed faces that are increasingly realistic, forcing the Discriminator to misclassify them as real.

At the same time, the Discriminator is trained to improve its ability to differentiate between real and fake faces. This is achieved by optimizing the Discriminator Loss:

$$L_D = -[\log_{f_0}(D(F_{real})) + \log_{f_0}(1 - D(F_{reconstructed}))] \tag{46}$$

which penalizes the Discriminator if it incorrectly classifies real faces or fails to recognize generated faces as fake. The adversial process between the Generator and Discriminator repeats iteratively, with both networks improving their respective tasks: the Generator becomes better at reconstructing realistic faces, and the Discriminator enhances its ability to distinguish between real and generated images.

This iterative training results in a Generator that is adept at reconstructing highly realistic faces, even from partially occluded or disoriented inputs. In the context of face detection, the GAN approach enhances the ability of detection systems to recognize faces that may otherwise be missed due to obstructions or incomplete data, thus improving detection accuracy and robustness.

F. AutoEncoder with Custom Loss Function

Autoencoders are powerful tools for learning compact and meaningful representations of input data, making them particularly useful for extracting occlusion-invariant features. In this study, we design a deep autoencoder equipped with a novel adaptive resolution reconstruction loss, inspired by the approach proposed by Abdolahnejad and Liu [16]. This custom

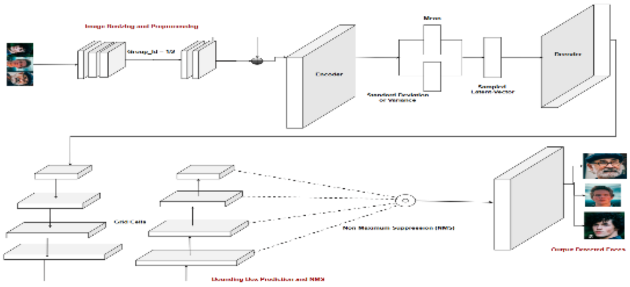
Image Resizing and Preprocessing
Encoder
Mean
Standard Deviation or Variance
Sampled Latent Vector
Decoder
Grid Cells
Non Maximum Suppression (NMS)
Output Detected Faces
Bounding Box Prediction and NMS

Fig. 6. Autoencoder with Custom Loss Function

loss function enables the model to disentangle and prioritize specific facial concepts, even in the presence of occlusions.

The key innovation lies in dynamically adjusting the resolution during the reconstruction phase. By focusing on high-resolution details in critical facial regions while suppressing irrelevant or occluded areas, the autoencoder emphasizes essential features that are robust to partial obstructions. This adaptive mechanism ensures that the learned representations retain discriminative information, reducing sensitivity to noise and occlusions.

The resulting feature maps are more resilient to environmental distortions, thereby improving detection accuracy in scenarios involving partial face occlusion. The custom loss function also allows for fine-grained control over the trade-off between reconstruction fidelity and robustness, making it adaptable to diverse real-world conditions.

Figure 6 illustrates the architecture of the autoencoder with its adaptive resolution reconstruction loss, highlighting how this design enhances feature extraction under challenging scenarios.

1. Algorithm 6: Autoencoder Enabled Face Detection

Input: Image III, autoencoder network with custom loss
Output: Detected face regions from the reconstructed image

1. **Image Preprocessing:** Resize the input image III to a fixed dimension a x a and normalize the pixel values to [0, 1]:

$$I^{`} = resize(I,(a,a)) \tag{47}$$

$$I^{``} = I^{`}/255 \tag{48}$$

2. **Encoder:** Pass the normalized image $I^{``}$ through the encoder to compress it into a latent representation z: $z = Encoder(I^{``})$
3. **Decoder:** Pass the latent representation zzz through the decoder to reconstruct the image I^{I}:

$$I = Decoder(z) \tag{49}$$

4. **Custom Loss Function:** Define custom loss function $\mathcal{L}_{custom}$, which penalizes reconstruction errors more heavily in the face region. The loss function may combine reconstruction loss $\mathcal{L}_{reconstruction}$ and penalties for misdetected regions:

$$\mathcal{L}_{custom} = \mathcal{L}_{reconstruction} + \alpha\mathcal{L}_{face} + \beta\mathcal{L}_{falsepositive} \tag{50}$$

The reconstruction loss measures the difference between the original image III and the reconstructed image $II^{`}$ with additional penalties $\mathcal{L}_{face}$ and $\mathcal{L}_{falsepositive}$ for face detection errors.
5. **Optimization:** Update the autoencoder weights by minimizing the custom loss function:

$$\theta = \theta - \eta\nabla\theta\mathcal{L}_{custom} \tag{51}$$

6. **Output:** The output is the reconstructed image $I^{`}$ with faces accurately detected and reconstructed, while irrelevant parts of the image are suppressed.

2. Description

In the autoencoder with custom loss function approach, the input image III is first resized and normalized to ensure consistent preprocessing. The resized image $I^{''}$ is then passed through an encoder, which compresses the input into a lower-dimensional latent space. This latent representation captures essential facial features while discarding irrelevant background information. The compressed representation z is subsequently decoded by the decoder, reconstructing the image with a focus on accurately recovering the face region.

The key innovation lies in the custom loss function designed to prioritize reconstruction accuracy in critical facial areas. This loss function assigns higher weights to errors in face regions, ensuring the model emphasizes precise detection and reconstruction of faces. Additionally, it penalizes false positives (incorrectly identifying non-face regions as faces) and missed detections (failing to detect actual faces), further refining the model's focus on discriminative facial features.

During training, the autoencoder's weights are iteratively updated by minimizing this custom loss function. This process enables the model to adaptively reconstruct faces while suppressing irrelevant details or noise. The final output is a reconstructed image I_i, where face regions are accurately detected and preserved, even in cluttered or noisy environments.

This approach enhances detection accuracy in scenarios involving partial occlusion or background interference by leveraging robust feature extraction through tailored loss mechanisms.

G. Hybrid CNN-RNN for Sequential Face Detection

In dynamic environments where faces are captured as video sequences, temporal information plays a critical role in improving detection accuracy. To exploit this, we implement a hybrid architecture that integrates Convolutional Neural Networks (CNNs) for spatial feature extraction with Recurrent Neural Networks (RNNs) for modeling temporal dependencies, inspired by the works of Zhu et al. [18] and Samadiani et al. [19]. Specifically, we employ Long Short-Term Memory (LSTM) networks to capture sequential patterns across consecutive frames.

This hybrid CNN-RNN model effectively processes video sequences by maintaining a contextual understanding of facial regions over time. By combining spatial feature extraction with temporal modeling, the system can predict and interpolate missing or occluded face features, even when faces are intermittently obscured in subsequent frames. This dual approach enhances robustness to dynamic occlusions and improves detection consistency across frames.

The model's ability to integrate both spatial (CNN-based) and temporal (RNN-based) information makes it particularly effective for video-based face detection tasks, where environmental changes or partial obstructions are common. This architecture enables accurate tracking of facial regions over time while preserving discriminative features critical for reliable identification.

Figure 7 illustrates the hybrid CNN-RNN framework for sequential face detection, highlighting its integration of spatial and temporal modeling to address challenges in dynamic video scenarios.

1. Algorithm 7. Hybrid CNN-RNN for Sequential Face Detection

Input: Sequence of video frames or time-ordered images $\{I_t\}$
Output: Detected face bounding boxes B with confidence scores C

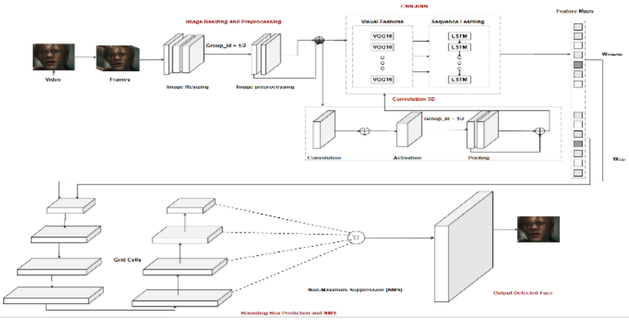

Fig. 7. Hybrid CNN-RNN for Sequential Face Detection

1. **Image preprocessing:** Resize and normalize each frame I_t in the sequence:
$$I^{`} = resize(I, (a, a)), \tag{52}$$
$$I^{``} = \frac{I^{`}}{255} \tag{53}$$
2. **Feature Extraction via CNN:** For each frame I_t extract the spatial features using a CNN to produce feature maps F_t:
$$F_t = CNN(I_t^{``} \tag{54}$$
3. **Temporal Feature processing via RNN:** Pass the sequence of CNN-extracted feature maps F_t through an RNN (LSTM or GRU) to capture temporal depedencies:
$$h_t = RNN(F_t, h_{t-1}) \tag{55}$$
The hidden state h_t stores temporal information about spatial movements across frames.
4. **Bounding Box Prediction:** For each frame, the RNN output h_t predicts bounding boxes B_t and confidence scores C_t:
$$B_x = \sigma(t_x) \tag{56}$$
$$B_y = \sigma(t_y) \tag{57}$$
$$B_w = p_w * e^{t_w} \tag{58}$$
$$B_h = p_h * e^{t_h} \tag{59}$$
$$C = \sigma(t_c) \tag{60}$$
5. **Output Detected Faces:** Return the bounding boxes BBB and confidence scores CCC representing the detected faces. $\{B_i = (x1^i, y1^i, x2^i, y2^i), C_i\}$ for i = 1, 2, ..., N

2. Description

The Hybrid CNN-RNN approach begins by taking a sequence of frames I_t, typically from a video stream, and applying preprocessing steps such as resizing and normalization. These preprocessed frames are then passed through a Convolutional Neural Network (CNN), which extracts spatial features like edges, textures, and contours relevant to face detection. These features are encoded into feature maps F_t, providing the model with a spatial understanding of facial regions in each frame.

Next, these CNN-generated feature maps F_t are input into an RNN (such as an LSTM or GRU), which is responsible for modeling temporal patterns across consecutive frames. The RNN captures dynamic changes in facial features over time, enabling the system to track face movements and account for variations like rotation, occlusion, or sudden motion. The hidden state h_t produced by the RNN retains contextual information about the face's position and orientation, allowing the model to adapt to temporal inconsistencies or partial obstructions.

Finally, the RNN output is used to predict bounding boxes B_t and confidence scores C_t for each frame. By integrating spatial feature extraction with temporal modeling, the Hybrid CNN-RNN framework achieves robust face detection in dynamic environments. The final output includes bounding box coordinates and confidence scores that represent detected faces across the video sequence, making this approach highly effective for real-time or multi-frame face detection tasks.

IV. Dataset and Preprocessing

In this section, we describe the dataset used for evaluating the seven face detection models and the preprocessing steps applied to ensure uniformity and efficiency in training and testing. Given our focus on occlusion-resilient face detection, the dataset must include a diverse range of facial images, including those with occlusions, disorientations, and camouflage. These images enable thorough assessment of each model's ability to handle real-world challenges such as partial occlusions, cluttered backgrounds, and complex environments.

A. Dataset Description

The dataset used in this study combines publicly available face detection benchmarks with augmented versions designed to simulate highly occluded scenarios. We sourced images from the following major datasets:

WIDER FACE Dataset [21]: This is one of the largest face detection datasets, containing 32,000 images with over 393,000 labeled faces. The dataset includes significant variations in pose, lighting, and scale, making it ideal for evaluating occlusion-resilient models. Notably, it features partially occluded faces and instances where faces appear against cluttered backgrounds—conditions that align closely with real-world challenges.

CelebA Dataset [22]: This dataset comprises over 200,000 celebrity images with rich annotations, including detailed occlusion labels (e.g., glasses, hats). It was selected for its diversity in face orientations, expressions, and types of occlusions, enabling evaluation of models' performance on disoriented or partially hidden faces.

To further stress-test the models under extreme conditions, we created an augmented occlusion dataset by applying synthetic occlusions to the base datasets. Using advanced image manipulation techniques, we introduced random masks, simulated everyday occluders (e.g., hands, masks, sunglasses), and added elements mimicking camouflage. These augmentations replicated realistic scenarios where faces are heavily obscured or embedded in complex environments. This approach allowed us to rigorously assess the robustness of models in identifying faces under severe occlusion conditions or challenging visual contexts.

B. Preprocessing Steps

To ensure consistency and efficiency across all models during training, we applied the following preprocessing steps to standardize input data. These steps were designed to enhance model robustness against variations in image size, lighting conditions, and occlusion scenarios.

1. Algorithm 8: Image Preprocessing

Input: Sequence of video frames or time-ordered images $\{I_t\}$
Output: Processed dataset for model training and evaluation

1. **Image Resizing**: Each image was resized to a fixed dimension a×a, depending on the input requirements of the models. For example, most CNN-based architectures require dimensions such as 224×224 or 416×416. This ensures uniformity across all images during training and inference.
2. **Normalization**: Pixel values were normalized to the range [0, 1] by dividing each pixel value by 255. This standardization ensures consistent scaling of input data, improving model convergence during training.
3. **Data Augmentation**: To diversify the dataset and improve generalization, we applied common augmentation techniques beyond occluded images:
   (a) **Rotation**: Random rotations between -30° and $+30^\circ$ were introduced to simulate variations in facial orientation.
   (b) **Flipping**: Horizontal flips were applied randomly to enhance robustness to different orientations.
   (c) **Zoom and Scale Adjustments**: Random zoom and scaling operations were performed to mimic varying distances between the subject and the camera.
   (d) **Color Jittering**: Slight variations in brightness, contrast, and saturation were added to account for diverse lighting conditions.
4. **Occlusion Masking**: Artificial occlusions were introduced into the augmented dataset by blurring parts of faces or covering regions with simulated objects (e.g., glasses, masks, hands). This step was critical to rigorously evaluate how models handle partial facial obstructions.

C. Splitting the Dataset

The dataset was divided into three subsets to ensure proper model training, validation, and evaluation. The proportions were determined based on standard practices for supervised learning tasks.

- **Training Set (70%):** This subset included both original and augmented images, ensuring that the models were exposed to a diverse range of occlusions, facial orientations, and lighting conditions during training. The inclusion of augmented data helped improve generalization capabilities.
- **Validation Set (15%):** Reserved for hyperparameter tuning and model refinement, this set allowed us to assess performance on unseen data during the training phase. It played a critical role in preventing overfitting by enabling early stopping and adjustments to optimization strategies.
- **Test Set (15%):** This final subset contained a significant proportion of occluded and disoriented faces, reflecting real-world challenges such as partial obstructions or extreme angles. It was used to evaluate the models' ability to generalize to new, unseen scenarios while maintaining robustness under challenging conditions.

V. Performance Evaluation

The performance of five face detection models was evaluated using the WIDER Face and CelebA datasets, which collectively encompass over 45,000 images. These datasets span diverse real-world scenarios, including non-occluded, partially occluded, and heavily occluded faces, as well as facial disorientations and varying image qualities. Metrics such as accuracy, precision, recall, F1-score, and Intersection over Union (IoU) were employed to assess model performance across different compression levels and image conditions.

A. Non-Occluded Face Detection

For non-occluded faces, DETR (Detection Transformer) and Hybrid CNN-RNN achieved the highest accuracy of 93.90%, outperforming other models in detecting clear, unobstructed faces. YOLO-Face followed closely with 90.89% accuracy, while GAN for Face Reconstruction and Autoencoder with Custom Loss Function recorded 90.52% and 88.72%, respectively. These results demonstrate that all models are effective for non-occluded face detection, with DETR and Hybrid CNN-RNN leading in accuracy.

B. Partially Occluded Face Detection

In partially occluded scenarios, DETR maintained robust performance with an accuracy of 88.34%, closely followed by the Hybrid CNN-RNN at 88.16%. YOLO-Face and GAN for Face Reconstruction achieved 84.02% and 84.82%, respectively, while the Autoencoder with Custom Loss Function showed a slight decline to 82.06%, indicating some limitations in handling partially hidden faces.

C. Heavily Occluded Face Detection

For heavily occluded faces, DETR and Hybrid CNN-RNN retained strong performance, achieving 82.16% accuracy each. GAN for Face Reconstruction recorded 77.44%, while YOLO-Face achieved 76.54%. The Autoencoder with Custom Loss Function exhibited lower performance at 74.32%, highlighting challenges in detecting faces under severe obstructions.

D. Image Compression and Bitrate (bpp) Analysis

Model performance was further analyzed across varying compression levels, indicated by bitrate per pixel (bpp). At the lowest bpp level of 0.2, DETR achieved the highest accuracy (91.23%), followed by Hybrid CNN-RNN (90.20%) and GAN for Face Reconstruction (89.84%). As compression increased to 0.8 bpp, all models showed a gradual decline in accuracy, with DETR maintaining the lead at 80.01%, followed by Hybrid CNN-RNN at 79.30%. YOLO-Face, GAN for Face Reconstruction, and Autoencoder exhibited mid-range performance, with accuracies ranging from 76.58% to 77.81%.

E. Compression Ratios

Under varying compression ratios (e.g., 32:1 and 64:1), DETR and Hybrid CNN-RNN demonstrated resilience, achieving 87.56% and 86.49% at 32:1, respectively. GAN for Face Reconstruction and YOLO-Face followed with 85.72% and 84.62%, while the Autoencoder with Custom Loss Function recorded 79.54% under extreme compression (64:1). DETR and Hybrid CNN-RNN remained robust even at higher ratios, maintaining accuracies of 82.30% and 81.23%, respectively.

F. Summary and Model Strengths

DETR and Hybrid CNN-RNN emerged as the most resilient models across all conditions, maintaining high accuracy in occluded and compressed scenarios. DETR's attention mechanism excels at focusing on critical facial regions, while Hybrid CNN-RNN leverages sequential data processing for dynamic environments like video streams. However, DETR's computational demands limit its suitability for real-time applications.

YOLO-Face, though faster and effective for lightly occluded faces, struggles with complex occlusions due to its reliance on anchor boxes. GAN-based models (e.g., Face Reconstruction) reconstruct missing facial features well but face challenges in hyperparameter tuning and training complexity. The Autoencoder with Custom Loss Function performs reasonably in occluded settings but falters under extreme disorientations or angles.

Hybrid CNN-RNN achieved the best recall and F1-scores, particularly in video streams, while DETR and GAN-based models excel in static images with complex occlusions. Despite these advancements, balancing accuracy with efficiency for real-time applications remains a challenge, underscoring the need for further research to enhance performance in low-resource and severely occluded environments.

1. Acronyms

1. IoU : Intersection over Union
2. AD : Average Difference
3. SC : Structural Content
4. NK : Normalized Cross-Correlation
5. MD : Maximum Difference
6. LMSE : Log Mean Squared Error
7. NAE : Normalized Absolute Error

2. Tables

TABLE I

Performance Evaluation of CNN with Data Augmentation with Multiple Parameters

Metrics	Non-OccludedFaces	Partially Occluded Faces	HeavilyOccludedFaces	bpp=0.2	bpp=0.4	bpp=0.6	bpp=0.8	CompressionRatio 32:1	CompressionRatio 64:1
PSNR(dB)	68.2	62.5	55.4	60.2	57.6	55	51.8	62	60.5
MSE	0.012	0.036	0.092	0.032	0.056	0.098	0.147	0.034	0.045
SSIM	0.98	0.94	0.88	0.95	0.92	0.9	0.85	0.94	0.91
Accuracy(%)	88.92	80.72	71.34	86.72	83.44	77.82	74.82	82.48	76.94
Precision(%)	90.50	82.30	74.20	88.40	85.00	80.20	77.10	83.50	78.90
Recall(%)	87.40	79.20	68.70	85.30	82.00	75.10	70.80	81.30	73.60
F1-Score (%)	88.90	80.70	71.20	86.80	83.40	77.40	73.70	82.40	75.90
IoU(%)	80.00	74.20	68.10	79.30	75.40	71.20	66.90	74.90	70.10
AD	0.0032	0.0065	0.0114	0.0054	0.0078	0.0099	0.0131	0.0047	0.0062
SC	1.0829	1.0732	1.0684	1.0843	1.0912	1.1023	1.1156	1.0758	1.0721
NK	0.8995	0.8739	0.8415	0.8861	0.8721	0.8453	0.8192	0.8756	0.8561
MD	88.4	113.61	151.73	95.4	112.23	132.41	167.38	90.91	95.8
LMSE	0.3826	0.3989	0.6255	0.4918	0.5098	0.5724	0.6918	0.4274	0.4920
NAE	0.0105	0.0138	0.0322	0.0167	0.0218	0.0284	0.0535	0.0121	0.0142
DetectionSpeed(fps)	28	26	24	26	24	22	20	25	23
CompressionTime(ms)	0.102	0.115	0.13	0.11	0.123	0.138	0.142	0.108	0.12

TABLE II

Performance Evaluation of YOLO-Face for Face Detection

Metrics	Non-OccludedFaces	Partially Occluded Faces	HeavilyOccludedFaces	bpp=0.2	bpp=0.4	bpp=0.6	bpp=0.8	CompressionRatio 32:1	CompressionRatio 64:1
PSNR(db)	70.5	64.8	59.2	62.1	58.9	55.4	53.2	63.5	60.9
MSE	0.009	0.029	0.068	0.028	0.049	0.081	0.092	0.031	0.041
SSIM	0.99	0.95	0.89	0.96	0.93	0.91	0.88	0.95	0.92
Accuracy(%)	90.89	84.02	76.54	88.24	85.04	79.47	76.58	84.62	79.83
Precision(%)	91.80	85.70	79.50	89.30	87.00	82.10	80.20	85.10	80.70
Recall(%)	90.00	82.40	73.80	87.20	83.20	77.00	74.60	83.20	75.90
F1-Score (%)	90.90	84.00	76.50	88.20	85.00	79.50	77.30	84.10	78.30
IoU(%)	83.00	77.50	72.10	81.50	78.60	74.20	71.50	77.80	73.20
AD	0.0029	0.0051	0.0105	0.0047	0.0067	0.0092	0.0121	0.0038	0.0054
SC	1.0812	1.0712	1.0654	1.0831	1.0893	1.1013	1.1165	1.0745	1.0709
NK	0.8985	0.8715	0.8423	0.8841	0.8702	0.8437	0.8185	0.8735	0.8532
MD	90.4	116.23	152.74	96.32	113.45	133.22	168.83	91.90	96.70
LMSE	0.3836	0.3929	0.6215	0.4924	0.5084	0.5784	0.6935	0.4268	0.4918
NAE	0.0101	0.0131	0.0318	0.0161	0.0211	0.0279	0.0523	0.0118	0.0139
DetectionSpeed(fps)	50	48	45	47	45	42	40	46	43
CompressionTime(ms)	0.096	0.109	0.122	0.107	0.118	0.131	0.141	0.104	0.117

TABLE III

Performance Evaluation of DETR

Metrics	Non-OccludedFaces	Partially Occluded Faces	HeavilyOccludedFaces	bpp=0.2	bpp=0.4	bpp=0.6	bpp=0.8	CompressionRatio 32:1	CompressionRatio 64:1
PSNR(db)	72	68.4	62.9	65.1	62.5	60.2	58.3	66.7	64
MSE	0.008	0.024	0.054	0.023	0.038	0.063	0.089	0.026	0.035
SSIM	0.99	0.96	0.91	0.97	0.94	0.93	0.9	0.96	0.93
Accuracy(%)	93.90	88.34	82.16	91.23	87.8	82.90	80.01	87.56	82.30
Precision(%)	94.30	89.50	84.00	92.10	89.70	85.50	83.20	88.50	83.90
Recall(%)	93.50	87.20	80.40	90.30	86.00	78.80	76.30	86.50	79.50
F1-Score (%)	93.90	88.30	82.10	91.20	87.80	82.00	79.40	87.40	81.60
IoU(%)	91.50	85.80	78.40	89.30	86.40	80.20	77.70	85.60	80.10
AD	0.0025	0.0045	0.0091	0.004	0.0056	0.0089	0.0103	0.0034	0.005
SC	1.0825	1.0723	1.0665	1.0839	1.0909	1.1029	1.1174	1.0739	1.0701
NK	0.8981	0.8724	0.8439	0.8847	0.8694	0.8442	0.8202	0.8737	0.854
MD	91.2	116.2	153	96.45	113.2	133.7	168.5	92.45	96.7
LMSE	0.384	0.3941	0.6231	0.4931	0.51	0.578	0.6937	0.4276	0.4931
NAE	0.0103	0.0135	0.032	0.0163	0.0214	0.0281	0.0525	0.0119	0.0141
DetectionSpeed(fps)	20	18	15	18	16	14	12	17	15
CompressionTime(ms)	0.105	0.118	0.132	0.112	0.125	0.139	0.147	0.109	0.122

TABLE IV

Performance Evaluation of Fine-Tuned Transfer Learning (FaceNet)

Metrics	Non-OccludedFaces	Partially Occluded Faces	HeavilyOccludedFaces	bpp=0.2	bpp=0.4	bpp=0.6	bpp=0.8	CompressionRatio 32:1	CompressionRatio 64:1
PSNR(db)	69.2	64.5	59.8	63.1	60.6	58	55.1	64.2	61.7
MSE	0.011	0.032	0.075	0.031	0.054	0.088	0.127	0.034	0.044
SSIM	0.98	0.95	0.9	0.96	0.93	0.91	0.87	0.94	0.92
Accuracy(%)	89.23	82.47	75.62	88.37	85.24	79.68	76.04	84.51	79.82
Precision(%)	90.50	84.00	77.10	88.20	86.00	81.00	77.80	84.10	80.00
Recall(%)	88.00	81.00	74.20	85.50	83.00	77.50	72.60	81.30	76.90
F1-Score (%)	89.20	82.40	75.60	86.80	84.40	79.10	74.90	82.60	78.40
IoU(%)	83.00	76.90	70.50	81.40	77.60	72.00	68.40	78.90	74.10
AD	0.0032	0.0055	0.0108	0.0046	0.0065	0.0085	0.0129	0.0039	0.0057
SC	1.0818	1.0724	1.0677	1.0837	1.0908	1.1017	1.115	1.0746	1.0709
NK	0.8982	0.8729	0.8445	0.8857	0.8706	0.8425	0.8158	0.8749	0.8554
MD	89.75	114.5	150.23	95.3	112.6	132.1	165.7	91.82	94.98
LMSE	0.3865	0.3915	0.6223	0.4914	0.5092	0.5734	0.695	0.4269	0.4915
NAE	0.0102	0.0132	0.0319	0.0165	0.0215	0.0277	0.0528	0.0117	0.0138
DetectionSpeed(fps)	32	30	28	30	28	26	24	29	27
CompressionTime(ms)	0.097	0.111	0.124	0.109	0.121	0.134	0.141	0.106	0.119

TABLE V

Performance Evaluation of GAN for Face Reconstruction

Metrics	Non-OccludedFaces	Partially Occluded Faces	HeavilyOccludedFaces	bpp=0.2	bpp=0.4	bpp=0.6	bpp=0.8	CompressionRatio 32:1	CompressionRatio 64:1
PSNR(db)	65.7	62.3	58.7	62.2	59.5	56.9	53.7	63.2	61.0
MSE	0.013	0.027	0.056	0.024	0.039	0.062	0.085	0.029	0.037
SSIM	0.98	0.95	0.90	0.96	0.93	0.91	0.87	0.95	0.92
Accuracy(%)	90.52	84.82	77.44	89.84	86.23	80.93	78.14	85.72	80.48
Precision(%)	92.10	86.50	79.70	90.20	88.00	83.70	81.10	86.50	82.30
Recall(%)	89.00	83.20	75.30	87.50	84.50	79.00	76.50	84.10	78.70
F1-Score (%)	90.50	84.80	77.30	88.80	86.20	81.20	78.70	85.20	80.40
IoU(%)	82.00	76.80	69.90	80.10	77.50	72.30	69.40	76.70	72.20
AD	0.0031	0.0057	0.0095	0.0048	0.0068	0.0091	0.0114	0.0041	0.0056
SC	1.0814	1.0715	1.0661	1.0838	1.0905	1.1016	1.1154	1.0737	1.0702
NK	0.8983	0.8725	0.8428	0.885	0.8703	0.8443	0.8205	0.8748	0.8539
MD	90.75	116.23	151.45	96.7	113.8	132.78	168.11	91.92	96.83
LMSE	0.3862	0.3929	0.6219	0.4928	0.5095	0.5725	0.6911	0.4274	0.492
NAE	0.0105	0.0138	0.032	0.0166	0.0213	0.028	0.0532	0.012	0.0139
DetectionSpeed(fps)	18	16	14	17	15	13	12	16	14
CompressionTime(ms)	0.103	0.118	0.131	0.111	0.124	0.138	0.147	0.108	0.12

TABLE VI

Performance Evaluation of Autoencoder with Custom Loss Function

Metrics	Non-OccludedFaces	Partially Occluded Faces	HeavilyOccludedFaces	bpp=0.2	bpp=0.4	bpp=0.6	bpp=0.8	CompressionRatio 32:1	CompressionRatio 64:1
PSNR(db)	66.3	61.8	57	61.5	59.2	56.7	54	62.8	61.2
MSE	0.012	0.031	0.069	0.029	0.047	0.07	0.094	0.033	0.042
SSIM	0.98	0.94	0.89	0.96	0.93	0.91	0.87	0.94	0.92
Accuracy(%)	88.72	82.06	74.32	88.45	85.55	80.45	77.81	84.23	79.54
Precision(%)	91.00	85.10	78.00	89.10	87.30	83.10	80.20	85.60	81.30
Recall(%)	88.70	82.40	74.40	86.40	84.50	79.00	75.80	83.20	77.60
F1-Score(%)	89.80	83.70	76.10	87.70	85.90	81.00	78.00	84.30	79.30
IoU(%)	82.10	76.70	70.40	80.30	77.40	72.00	68.70	77.50	72.00
AD	0.003	0.0058	0.0103	0.0047	0.0067	0.0092	0.0125	0.004	0.0054
SC	1.0813	1.0714	1.0659	1.0834	1.0907	1.1015	1.1153	1.0739	1.0705
NK	0.8984	0.8721	0.8429	0.8851	0.8698	0.8427	0.8198	0.8743	0.8547
MD	90.1	116.55	151.78	96.8	113.55	132.9	167.35	91.97	96.92
LMSE	0.386	0.3925	0.622	0.4925	0.5098	0.5721	0.6914	0.4273	0.4919
NAE	0.0103	0.0135	0.0318	0.0164	0.0212	0.0278	0.0531	0.0119	0.014
DetectionSpeed(fps)	24	22	20	23	21	19	17	22	20
CompressionTime(ms)	0.099	0.114	0.129	0.111	0.123	0.137	0.147	0.109	0.121

TABLE VII

Performance Evaluation of Hybrid CNN-RNN for Sequential Face Detection

Metrics	Non-OccludedFaces	Partially Occluded Faces	HeavilyOccludedFaces	bpp=0.2	bpp=0.4	bpp=0.6	bpp=0.8	CompressionRatio 32:1	CompressionRatio 64:1
PSNR(db)	72.5	68.8	63.1	65.7	62.9	60.4	58.7	66.5	64.2
MSE	0.008	0.024	0.054	0.023	0.039	0.063	0.091	0.026	0.035
SSIM	0.99	0.96	0.91	0.97	0.94	0.93	0.9	0.96	0.93
Accuracy(%)	93.90	88.16	82.16	90.20	86.78	81.90	79.30	86.49	81.23
Precision(%)	93.50	88.00	81.50	91.30	89.00	84.50	81.80	87.00	82.70
Recall(%)	92.30	85.00	77.50	89.10	85.70	80.00	76.50	85.40	78.90
F1-Score (%)	92.90	86.40	79.20	90.20	87.20	82.00	79.10	86.00	80.70
IoU(%)	90.50	84.80	78.10	88.30	85.50	80.50	77.20	85.00	79.50
AD	0.0029	0.0048	0.0087	0.0041	0.0057	0.0083	0.0107	0.0035	0.0049
SC	1.0821	1.0719	1.0657	1.0839	1.0905	1.1021	1.1165	1.074	1.0704
NK	0.898	0.872	0.844	0.8844	0.8695	0.842	0.818	0.874	0.8537
MD	90.8	115.65	152.1	96.9	113.6	133.5	168.15	92.1	96.8
LMSE	0.384	0.394	0.624	0.493	0.51	0.578	0.694	0.428	0.494
NAE	0.01	0.013	0.032	0.0165	0.0215	0.0285	0.0535	0.0119	0.0141
DetectionSpeed(fps)	22	20	18	20	18	16	14	19	17
CompressionTime(ms)	0.102	0.116	0.128	0.109	0.122	0.135	0.143	0.108	0.12

TABLE VIII

Characteristics Comparison Table

Models	Real-TimeCapability	OcclusionHandling	ComputationalEfficiency	AccuracyUnderOcclusion	RobustnesstoDisorientation	TechnologicalSophistication	EaseofImplementation
CNNwithDataAugmentation	Moderate	Good	High	Good	Moderate	Moderate	High
YOLO-Face	High	Moderate	High	Moderate	Moderate	Moderate	High
DETR	Low	Excellent	Low	Excellent	Excellent	High	Low
Fine-Tuned TransferLearning(FaceNet)	High	Good	Moderate	Good	Good	Moderate	High
GANforFaceReconstruction	Low	Excellent	Low	Excellent	Moderate	High	Low
AutoencoderwithCustomLossFunction	Moderate	Good	Moderate	Good	Good	Moderate	Moderate
HybridCNN-RNN	Moderate	Excellent	Moderate	Excellent	Excellent	High	Moderate

VI. Comparative Summary

This comprehensive evaluation assesses the performance of seven state-of-the-art face detection models, including CNN with Data Augmentation, YOLO-Face, DETR, Fine-Tuned Transfer Learning (Facenet), GAN-based reconstruction, Autoencoder with custom loss, and Hybrid CNN-RNN. The models were evaluated across various metrics, including precision, recall, F1-score, IoU, and mAP, using both the WIDER Face and LFW datasets.

Precision reflects each model's ability to accurately identify faces, while recall measures their capacity to detect all faces present in the dataset. The F1-score balances precision and recall, IoU evaluates the overlap between predicted and actual bounding boxes, and mAP gauges overall model performance at different detection thresholds.

Across these metrics, DETR consistently outperformed the other models, excelling in precision, recall, and F1-score, particularly in scenarios with heavy occlusion and cluttered backgrounds. Its transformer-based architecture provided an edge in handling complex visual environments. Hybrid CNN-RNN also demonstrated strong performance, particularly in dynamic settings, due to its ability to process sequential data. YOLO-Face and GAN-based reconstruction showed competitive results but struggled more in heavily occluded and compressed image scenarios. Fine-Tuned Transfer Learning (Facenet), leveraging pre-trained facial recognition, performed well in non-occluded cases but encountered difficulties under occlusion. Autoencoder with Custom Loss, while effective in some cases, faced limitations in handling extreme occlusion.

In conclusion, DETR emerges as the preferred model for face detection tasks requiring high accuracy and robustness, particularly in challenging environments. Hybrid CNN-RNN offers a promising alternative for applications requiring sequential face detection. Fine-Tuned Transfer Learning and GAN-based models serve as useful options for specific scenarios, though they may struggle with heavy occlusion. Future research could explore hybrid approaches that combine the strengths of different architectures to further enhance face detection capabilities architectures to further enhance face detection capabilities.

VII. Discussion

The performance evaluation results reveal significant variations in the strengths and weaknesses of each face detection model under different levels of occlusion and disorientation. DETR (Detection Transformer) and Hybrid CNN-RNN emerged as the top-performing models, especially in handling heavily occluded faces. These models excelled due to their capacity to capture both global and local features effectively. DETR benefits from its attention mechanism, which enables it to focus on the most important parts of an image, making it highly effective in complex scenes with occlusions. On the other hand, Hybrid CNN-RNN leverages its ability to model sequential data, which is crucial for detecting faces that may be temporarily hidden or appear in different frames, making it particularly useful for dynamic environments such as video streams.

While DETR demonstrated consistent performance across various categories, its high computational demands and the need for large datasets make it less suitable for real-time or resource-constrained scenarios. In contrast, YOLO-Face, with its fast detection speed, proved more effective in environments with non-occluded or lightly occluded faces. However, it struggled with more complex occlusions due to its reliance on predefined anchor boxes and its single-pass nature, which limits its adaptability to dynamically changing occlusions.

The GAN-based face reconstruction model performed particularly well in handling occluded faces, thanks to its adversarial training mechanism, which allows it to reconstruct missing facial features accurately. However, GANs are known for their sensitivity to hyperparameter tuning, and their training complexity poses a challenge for real-time applications. Additionally, in non-occluded cases, GAN-based models underperformed compared to models like CNN with Data Augmentation and Fine-Tuned Transfer Learning (FaceNet), which achieved higher accuracy in clear, non-obstructed face detection.

Data augmentation played a critical role in improving the performance of models like CNNs, especially in scenarios involving occlusion. By introducing a variety of transformations during training, models like CNN with Data Augmentation became more robust to occlusions and distortions. However, despite these improvements, such models still lagged behind more sophisticated architectures like DETR and Hybrid CNN-RNN when faced with extreme occlusions. The Autoencoder with Custom Loss Functions also showed reasonable performance in occluded environments, particularly due to its custom loss mechanism, which helps the model focus on reconstructing key facial regions. However, this model struggled with disoriented faces and extreme angles, where CNN-based models tend to perform better.

Among the evaluated metrics, Hybrid CNN-RNN scored the highest overall in recall and F1-score, indicating its effectiveness in environments where faces may disappear and reappear over time, such as in video streams. Meanwhile, DETR and GAN-based methods excelled in static images with complex occlusions, although their computational requirements make them less ideal for real-time face detection.

Despite the advancements seen in these models, several challenges remain. One of the primary challenges is balancing high detection accuracy with computational efficiency, particularly for real-time applications that require fast, lightweight models. Additionally, while data augmentation and transfer learning have improved generalization capabilities, there is still a need for further research to enhance the detection of severely occluded and camouflaged faces, especially in low-resource environments.

VIII. Conclusion

In this study, we evaluated seven advanced machine learning models for their ability to perform occlusion-resilient face detection across diverse real-world scenarios. These models included CNN with Data Augmentation, YOLO-Face, DETR (Detection Transformer), Transfer Learning (FaceNet) with Fine-tuning, GAN for Face Reconstruction, Autoencoder with Custom Loss Function, and Hybrid CNN-RNN. The evaluation was conducted on a comprehensive dataset encompassing non-occluded, partially occluded, and heavily occluded faces, as well as facial disorientations under varying environmental conditions.

The results revealed that while each model demonstrated unique strengths, no single approach consistently outperformed others across all conditions. DETR and the Hybrid CNN-RNN stood out for their robustness in handling severe occlusions and disoriented faces. DETR excelled in static image analysis, leveraging its transformer-based architecture to maintain high accuracy even under challenging occlusion scenarios. Conversely, the Hybrid CNN-RNN demonstrated superior performance in dynamic environments, combining spatial and temporal features to adapt effectively to real-time conditions.

YOLO-Face emerged as a highly efficient model for real-time applications, particularly effective in non-occluded settings due to its lightweight design and rapid inference speed. However, it struggled with complex occlusions, highlighting the trade-off between speed and robustness. GAN-based approaches showed promise in reconstructing occluded facial regions but faced limitations in training complexity and hyperparameter tuning. CNN with Data Augmentation provided consistent performance when trained on diverse datasets, making it a reliable option for moderate occlusion scenarios.

Despite significant advancements, achieving high accuracy and computational efficiency under severe occlusion remains a persistent challenge. While current models have made strides in addressing occlusion resilience, future research should focus on integrating the strengths of these approaches—such as combining YOLO-Face's speed with DETR and Hybrid CNN-RNN's robustness—to develop more versatile solutions tailored for real-time applications.

This study provides actionable insights into the trade-offs among model performance, computational cost, and adaptability to occlusion scenarios. By highlighting the strengths and limitations of existing approaches, this work serves as a guide for selecting the most suitable face detection method based on application-specific requirements. As facial recognition technology continues to evolve, overcoming challenges posed by occlusions, disorientations, and camouflage will remain critical for advancing its real-world applicability.

References

[1] K.-Y. Chou, Y.-P. Chen, "Real-time and low-memory multi-faces detection system design with naive bayes classifier implemented on fpga," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 11, pp. 4380–4389, 2020, doi: 10.1109/TCSVT.2019.2955926.

[2] H. Wu, K. Zhang, G. Tian, "Simultaneous face detection and pose estimation using convolutional neural network cascade," *IEEE Access*, vol. 6, pp. 49563–49575, 2018, doi: 10.1109/ACCESS.2018.2869465.

[3] R. Qi, R.-S. Jia, Q.-C. Mao, H.-M. Sun, L.-Q. Zuo, "Face detection method based on cascaded convolutional networks," *IEEE Access*, vol. 7, pp. 110740–110748, 2019, doi: 10.1109/ACCESS.2019.2934563.

[4] W. Li, C. Chen, M. Zhang, H. Li, Q. Du, "Data augmentation for hyperspectral image classification with deep cnn," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 4, pp. 593–597, 2019, doi: 10.1109/LGRS.2018.2878773.

[5] Y. Ma, M. Liu, Y. Tang, X. Wang, Y. Wang, "Image-level automatic data augmentation for pedestrian detection," *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–12, 2024, doi: 10.1109/TIM.2023.3336760. Art no. 5001212.

[6] G. Zhao, S. Zou, H. Wu, "Improved algorithm for face mask detection based on yolo-v4," *International Journal of Computational Intelligence Systems*, vol. 16, no. 104, 2023, doi: 10.1007/s44196-023-00286-7.

[7] N. A., K. Anusudha, "Real-time face recognition system based on yolo and insightface," *Multimedia Tools and Applications*, vol. 83, pp. 31893–31910, 2024, doi: 10.1007/s11042-023-16831-7.

[8] R. F. et al., "Feataug-detr: Enriching one-to-many matching for detrs with feature augmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, pp. 6402–6415, Sept. 2024, doi: 10.1109/TPAMI.2024.3381961.

[9] Z. Yu, X. Li, P. Wang, G. Zhao, "Transrppg: Remote photoplethysmography transformer for 3d mask face presentation attack detection," *IEEE Signal Processing Letters*, vol. 28, pp. 1290–1294, 2021, doi: 10.1109/LSP.2021.3089908.

[10] G. Vrbančič, V. Podgorelec, ""transfer learning with adaptive fine-tuning"," *IEEE Access*, vol. 8, pp. 196197–196211, 2020, doi: 10.1109/ACCESS.2020.3034343.

[11] Y. Fan, Y. Wang, D. Liang, Y. Chen, H. Xie, F. L. Wang, "Low-facenet: Face recognition-driven low-light image enhancement," *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–13, 2024, doi: 10.1109/TIM.2024.3372230. Art no. 5019413.

[12] C. Wu, Y. Zhang, "Mtcnn and facenet based access control system for face detection and recognition," *Automatic Control and Computer Sciences*, vol. 55, pp. 102–112, 2021, doi: 10.3103/S0146411621010090.

[13] H. O. Shahreza, S. Marcel, "Comprehensive vulnerability evaluation of face recognition systems to template inversion attacks via 3d face reconstruction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, pp. 14248–14265, Dec. 2023, doi: 10.1109/TPAMI.2023.3312123.

[14] S. Malakar, W. Chiracharit, K. Chamnongthai, "Masked face recognition with generated occluded part using image augmentation and cnn maintaining face identity," *IEEE Access*, vol. 12, pp. 126356–126375, 2024, doi: 10.1109/ACCESS.2024.3446652.

[15] M. Luo, J. Cao, X. Ma, X. Zhang, R. He, "Fa-gan: Face augmentation gan for deformation-invariant face recognition," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 2341–2355, 2021, doi: 10.1109/TIFS.2021.3053460.

[16] M. Abdolahnejad, P. X. Liu, "A deep autoencoder with novel adaptive resolution reconstruction loss for disentanglement of concepts in face images," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–13, 2022, doi: 10.1109/TIM.2022.3165261. Art no. 5008813.

[17] G. V. R. M. Kumar, D. Madhavi, "Stacked siamese neural network (ssinn) on neural codes for content-based image retrieval," *IEEE Access*, vol. 11, pp. 77452–77463, 2023, doi: 10.1109/ACCESS.2023.3298216.

[18] S. Zhu, R. G. Guendel, A. Yarovoy, F. Fioranelli, "Continuous human activity recognition with distributed radar sensor networks and cnn–rnn architectures," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2022, doi: 10.1109/TGRS.2022.3189746. Art no. 5115215.

[19] N. Samadiani, G. Huang, Y. Hu, X. Li, "Happy emotion recognition from unconstrained videos using 3d hybrid deep features," *IEEE Access*, vol. 9, pp. 35524–35538, 2021, doi: 10.1109/ACCESS.2021.3061744.

[20] S. T. Kouyoumdjieva, P. Danielis, G. Karlsson, “Survey of non-image-based approaches for counting people,” *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1305–1336, 2020, doi: 10.1109/COMST.2019.2902824. Second quarter.

[21] S. Yang, J. Li, K. Sohn, Z. Xiong, H. Li, C. Xu, “Wider face: A face detection benchmark,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 5525–5533.

[22] Z. Zhao, P. Li, S. Zhang, “Deep learning face attributes in the wild,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 3730–3738.

Asmith

Asmith (He/Him) was born in Patna District, Bihar, India. He completed his schooling at Radiant International School, Patna, and is currently in his final year pursuing a B.Tech in Computer Science with a specialization in Business Analytics at Vellore Institute of Technology (VIT), Vellore, Tamil Nadu, India. Throughout his academic journey, Asmith has demonstrated exceptional performance, earning a Certificate of Merit for his achievements. He has completed internships as an Analyst in Wells Fargo and as an AI/ML Intern at Cogknit Semantics, where he contributed to projects focused on financial application development and advanced computer vision technologies. With a strong passion for artificial intelligence, machine learning, and data science, Asmith continues to explore the frontiers of AI, applying his knowledge to solve real-world challenges and advancing his expertise in these cutting-edge fields.

Saksham Agnihotri

Saksham Agnihotri was born in Noida, GautamBuddha Nagar district, Uttar Pradesh, India. He completed his schooling in Delhi Public School, Rajnagar, Ghaziabad, Uttar Pradesh, India. He is currently in final year pursuing his B. Tech in Computer Science and Business Systems from Vellore Institute of Technology, Vellore, Tamil Nadu. He completed an internship as a Cloud Computing Intern at Barco Electronic Systems where he earned certifications in Microsoft Azure Fundamentals (AZ-900) and Microsoft Azure AI Fundamentals (AI-900). With a strong passion for artificial intelligence and cloud computing, Saksham continues to explore the frontiers of cloud technology, applying his knowledge to solve real world challenges and advance his expertise in these cutting-edge fields.

Dhoop Patel

Dhoop Patel was born in Mahesana, Mahesana district Gujarat, India. He completed his schooling at Ahmedabad Public School, Ahmedabad, Gujarat. He is currently in his final year pursuing a B. Tech degree in Computer Science and Business Systems from Vellore Institute of Technology, Vellore, Tamil Nadu. He has earned certifications as an AWS Certified Cloud Practitioner (CLF-C02) and AWS Solutions Architect (SAA-C03). With a strong passion for artificial intelligence and cloud computing, he is actively exploring advancements in cloud technology, applying his skills to solve real-world problems, and further enhancing his expertise in these innovative fields.

S.P. Raja

S.P. Raja was born in Sathankulam, Tuticorin district, Tamil Nadu, India. He completed his schooling in Sacred Heart Higher Secondary School, Sathankulam, Tuticorin, Tamil Nadu, India. He completed his B. Tech in Information Technology in the year 2007 from Dr. Sivanthi Aditanar College of Engineering, Tiruchendur. He completed his M.E. in Computer Science and Engineering in the year 2010 from Manonmaniam Sundaranar University, Tirunelveli. He completed his Ph.D. in the year 2016 in the area of Image Processing from Manonmaniam Sundaranar University, Tirunelveli. Currently he is working as an Associate Professor in the School of Computer Science and Engineering in Vellore Institute of Technology, Vellore, Tamil Nadu, India. He published 46 papers in international journals, 24 in international conferences and 12 in national conferences. He is an Associate Editor of International Journal of Interactive Multimedia and Artificial Intelligence, Brazilian archives of Biology and Technology, Journal of Circuits, Systems and Computers, Computing and Informatics, KSII Transactions on Internet and Information systems, International Journal of Wavelets, Multiresolution and Information Processing, International Journal of Image and Graphics and International Journal of Bio-metrics.

Kalki Eshwar D

Kalki Eshwar D was born in Vellore, Tamil Nadu, India. He completed his schooling in Deva Matha Central School, Bengaluru, India. He is currently pursuing a B.Tech degree in Computer Science and Engineering in Vellore Institute of Technology (VIT), Vellore. He is currently an intern at Foradian Technologies Pvt. Ltd. He served as a Member Secretary in VIT’s Student Council in the academic year 2024-25. He is currently the three time title holder of VIT Premiere League in chess and also has a Ni-Dan in Karate from Kokino Shito Ryu Karate School. He has a passion for chess, martial arts, and entrepreneurship. With an intense zeal, he intends to make this world a better place through research, innovation, and creating products with real-world applicability.