Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 12 , December - 2025

A Survey on Components of an End-to-End Face Detection System: Algorithms, Limitations and Intelligent Computing

Snehal D. Patil
Electronics and Telecommunication
COEP Tech. University
Pune, India

Dr. Prashant P. Bartakke
Electronics and Telecommunication
COEP Tech. University
Pune, India

Dr. Mukul S. Sutaone
Electronics and Telecommunication
IIIT Allahabad
Allahabad, India

Mr. Rajesh Chavan
Research and Development
Fourfront Pvt. Ltd
Pune, India

Abstract - **Face detection is the very first step for an efficient face recognition system. The success of an application for face recognition is hinged rigidly with the implementation of a coherent face detection system. Before designing an end-to-end system, foremost attention must be provided in choosing a desired face detection technique based on system architecture, complexity, inference time and limitations it offers. Most of the real-world applications like home automation, attendance, tourism, banking, security, automobile, immigration, retail, healthcare expects a prompt response, making it essential that inference time from each module of an application should be optimized. Face detection is constrained by several challenges that consists of occlusion, illumination, fake face, scale, pose, low resolution, make-up, extreme expressions, reflection, complexion etc. The emergence of cloud and edge computing platforms enabled to meet the huge deficit among training data and computation resources. The deployment of the model on embedded platform is often a concern due to enormous model size. In this survey exhaustive analysis of various face detection techniques is put through regarding network architecture, limitations, python packages, performance metrics and advantages. Also, the components of an end-to-end application development imbibing face detection is accomplished. Custom face detection models hunt effectively for the existing constraints and uncontrolled conditions. Integration of face detection systems and Advanced Driver Assistance Systems (ADAS) platforms on cloud instances is consummated to cater for emerging need of security in automotive sector. Towards the end, a brief overview of the challenges in this field and dimensions for future research are contemplated.**

face detection, driver monitoring system, intelligent computing, cloud instances, PYPI packages

I. INTRODUCTION

Face Localization is the foundation for computer vision tasks. Intelligent computing is the buzz word in advanced driver assistance systems (ADAS), surveillance and driver monitoring systems (DMS). All these systems comprise of a video capturing camera and an analysis system to initiate a proactive action for prevention of hazard. Deep Learning is a boon sustaining scalability of the data collection devices and enhancing the computing abilities to take a decision accurately. Face Detection is affected due to different factors such as profile view, styling accessories, fake images, scale, and cluttered background. Face Localization is the specific case of object detection task. The entire journey towards making life simpler started with the emergence of technologies imitating human thinking. The advanced computing techniques are modifying the dimensions in which data can be interpreted and analyzed. The inter relation among the above techniques and the way these techniques handle data is depicted (refer Figure 1). Artificial intelligence is the capability of computers to acquire information from the input data. It solves problems effectively by devising optimal and adaptive techniques without human intervention. The origin of Intelligent systems can be traced back with a novel question “Can machines think?” [1]. If the evaluator is unable to distinguish the replies between the person and a general-purpose computer in an imitation game, the processor is declared to be the winner. The goal is straight forward to persuade the assessor about their interaction with a human being rather than an intelligent device. Machine Learning has been introduced to make our life easier. Early intelligent systems used programmed conditional statements to analyze the user information. But with Machine Learning, data is made available to learn discriminative features and understand patterns from it. Machine Learning detour the need to undergo repetitive coding for every new query encountered unlike accustomed issues, the similar method needs to be used with different dataset. Deep Learning utilizes computational techniques and experiential data for training purposes inspired by our brain’s own network of neurons. It interprets the information that acts as an input and process the information from cumulative experiences. The base line for Deep Learning algorithms derives from experience gathering and active learn-

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 12 , December - 2025

ing. Computers learn through a network of neurons and boost their properties without specific programming or mathematical modeling.

Deep Learning is gaining significant attention in various application domains. The detailed study of face detections commences with the conventional methods, moving towards machine learning approaches and settles down on deep learning techniques. Processing enormous amounts of data imposes restrictions on computing resources. To surpass this challenge, cloud and edge computing proves effective. Comprehensive discussion of different cloud instances and machine learning ADAS platforms is performed. Cloud instances such as AWS Sagemaker, Azure machine learning, IBM SoftLayer, and Google Cloud AutoML empower model development and deployment allowing cross framework interoperability and GUI development tools. NVDIA, NXP, and AWS are promoting the design of driver monitoring systems featuring multi framework and inference engine support. System architecture, parallel computing, hardware, and software acceleration are the key aspects affecting the performance of programming devices. The first step towards an application development is selection of the system model with consideration of model size, computing resources, training, and inference time. Graphic processing unit (GPU) is an alternative solution to quench the limited processing abilities of a general-purpose computer during training. Compute Unified Device Architecture from NVIDIA is a parallel processing GPU platform enabling computation intensive tasks of an application to be dispatched concurrently. Field programmable gate array (FPGA) has an added advantage of hardware reconfiguration and model optimization. To enable real time inferencing, deployment of trained model on embedded platform is significant. The model configuration is gigantic making direct flashing on end device less feasible. The amalgamation of various components lays down firm pillars for a competitive application. This survey provides literature in emerging intelligent computing facilities, python packages, network architecture, advantages and flaws, factors for inaccurate detection, loss functions, performance metrics, dataset for face detection application, cloud computing platforms, machine learning based advanced driver assistance systems. For the organization of the paper and subsections corresponding to it (refer Figure 2)

II. EVOLUTION OF FACE DETECTION METHODS

A. Image Processing face detection methods

Face processing is gaining attention in the arena of computer vision domain owing to uniqueness of an individual face. Human face detection is a significant research area incorporating diverse applications like video surveillance, access control, face verification, face expression, advanced computer, and human interaction. Early face localization can be roughly categorized into local facial features detection, template matching and image invariants. Conventional face detection systems originally were based on facial feature excerption utilizing low level computer vision methods and classification based on statistical models [2] [3] [4] [5]. Template matching comprising several correlation templates are used to detect local sub features [6]. Wavelet packet decomposition encompassing skin colour sieving and likelihood categorization of facial textures was widely used for rapid face detection. In case of image invariants, there is an assumption of spatial image relationships unique to all image patterns and under various imaging conditions [7]. The early methods have drawback of limited global constraints applied on face templates and the features extracted are greatly affected by noise or expression or viewpoint. Correlation based template matching methods are computationally expensive and require a large amount of storage. Alternative approaches for human face detection instead of handcrafted features are based on neural networks wherein [8] [9] ample information is provided to learn and identify patterns in data.

B. Machine Learning face detection methods

In these methods handcrafted features are extracted. Although the methods are popular and widely used, it has limitations in terms of computation time and detection of false positives.

1) Haar Cascade or Viola Jones: Several techniques are proposed and widely used for Face Detection, but the foremost and successful algorithm widely used was in 2001 by Viola and Jones. They proposed a framework for object detection in real time from video footage. This algorithm was proposed long back when Deep Learning had not been even given thought off. There are two approaches for seeking information from an image feature based and pixel based. The advantage of using feature-based systems is that they're faster as compared to pixel-based systems. In the above figure, the darker areas represent pixel value 1 and lighter areas represent pixel value 0. Haar features are used for finding features in an image. Feature extraction is dominated by sharp intensity variations across a line or any other arrangement. In order to extract vertical features with low intensity pixels on the right side and high intensity ones on the left side (refer Figure 3A) Also, to procure horizontal features with low intensity pixels above the edge and low intensity pixels below it (refer Figure 3B). The purpose is to calculate the sum of all pixel intensities in the darker area and the sum of all pixel intensities in the lighter area of the haar feature. Now if there is an edge appearing between darker and lighter pixels, then feature value is evaluated approximately near to 1. This mathematical process is carried out to synthesize different features from image or video under consideration.

2) Histogram of Oriented Gradients (HoG): It is robust as compared to haar cascades classifier and works effectively in different illumination conditions. For workflow of HoG (refer Figure 4). This technique is based on counting the gradients in a localized portion of an image. Feature detection principle works on transforming from image space to parameter space. Image space consists of Cartesian coordinates as parameter while parameter space consists of slope and y intercept as parameter. So, point in image space maps to line in parameter

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 12 , December - 2025

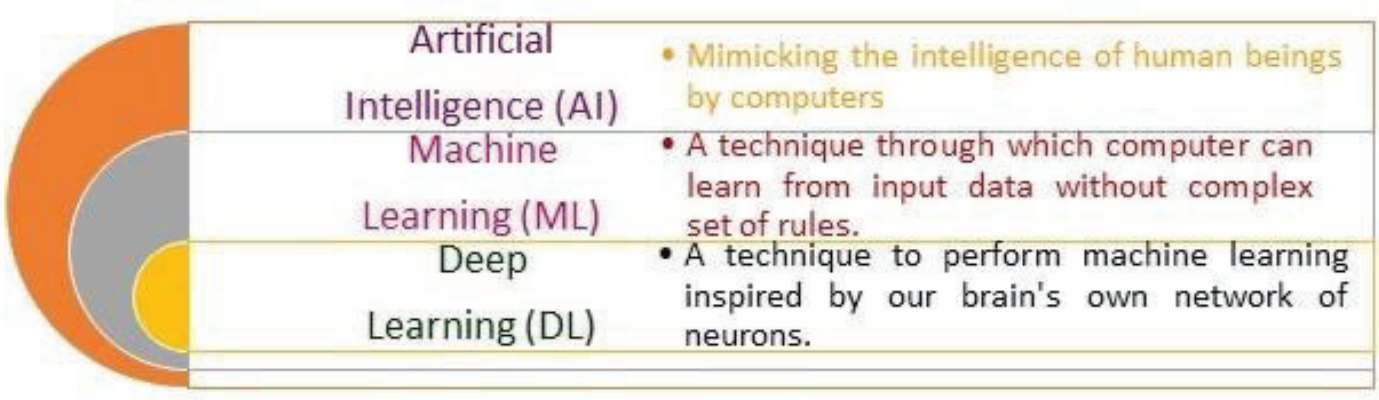

Fig. 1. Inter Relation among intelligent computing systems- artificial intelligence, machine learning, and deep learning

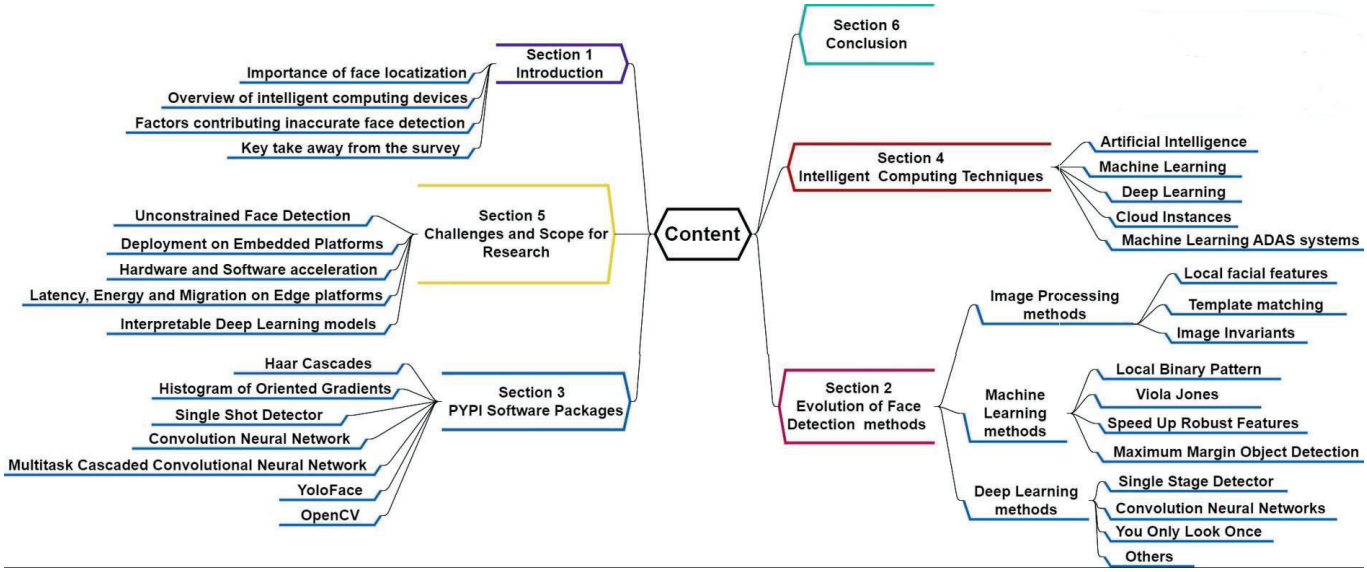

Fig. 2. Structure of the Survey

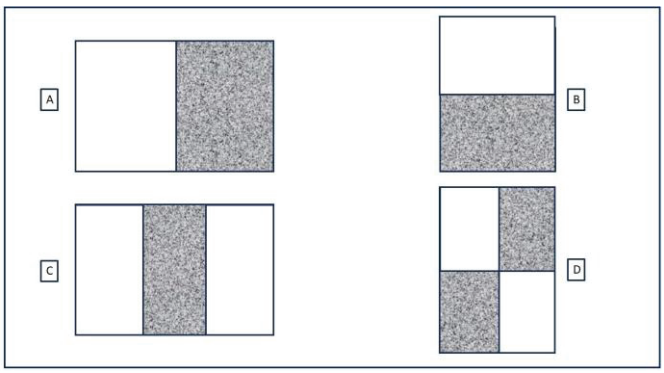

Fig. 3. Haar Feature Extractor

space. As the value of slope ranges from minus infinity to plus infinity, there is limitation that parameter space requires large size of accumulator and memory computations. The solution for this is to reconfigure the parameter space in terms of distance from origin and angle. As both distance and angle are finite, the size of accumulator reduces drastically. The calculated gradients are further processed to evaluate the histograms that are further converted to HoG description vectors and classified with the aid of Support vector machine (SVM) classifier. Viola Jones is a preliminary machine learning approach for rapid processing of images with high detection

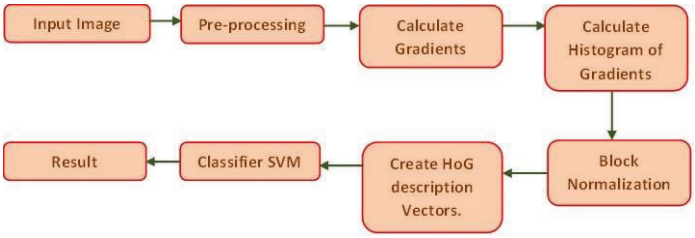

Fig. 4. Architecture of Histogram of Oriented Gradients

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 12 , December - 2025

accuracy indexing huge amount of video and image data. The three major contributions that distinguish this methodology from others are Integral Image, small critical visual features with Adaboost learning algorithm, combination of complex classifiers in cascade focusing on promising object regions and eliminating background regions [10]. Deciphering the ingredients of first ever real time face detection system was illustrated in [11]. Multiple redundant detections from Viola-Jones algorithm is conquered with a robustness argument in the past processing step. Speed up Robust Features (SURF) [12] is an extension work of approach yielding much faster training convergence with AUC as the single convergence criterion as against Viola Jones framework which utilizes two inconsistent metrics (sensitivity and discerning rate). Human facial factors such as eyes, nose, mouth, and face are detected with haar cascade object detector [13]. Li et al. [14] proposed a system of haar cascades with three additional classifiers to get rid of non-human faces. An improved Haar cascades algorithm for face localization with Microsoft HoloLens achieved 12% on average higher detection efficiency and four-fold detection speed than that of existing approach [15]. HOG with SVM utilizes a classifier that constructs high dimensional concatenated HOG features. The high dimensional representation not only results in time consuming training procedures but also leads to slow detection speed. Human detection system based on HOG is implemented in [16] [17] [18]. HOG can perform better when learned via MMOD (max margin object detection) [19]. Partially overlapping windows with objects are difficult to imbibe into training set as these are neither a false alarm nor true detection. MMOD leads to convex optimization by working on missed detections and false alarms. The next logical step after an efficient face detection model is human detection. Local Binary Pattern LBP [20] has shown effectiveness in diverse tasks of face recognition, facial detection, facial expression analysis and demographic classification. LBP is a non-parametric descriptor summarizing the local structures in an image by comparing each pixel with its neighboring one. Conventional methods accomplished significant detection performance on existing approaches prevelant at that time [9] [8]. Face localization encounters two major challenges: a. Enormous visual changes in bestrewed backgrounds b. huge search area for diverse face sizes. These challenges impose a time efficiency requirement and effective binary classifier. Deep Learning significantly improves training time, resources utilization and accuracy.

C. Deep Learning Based Methods

Deep Learning approach handles enormous amount of data for taking appropriate decision. The tradeoff between accuracy and computation needs to be managed with utmost care. The architecture of SSD is as shown in (refer Figure 5). Table I shows comparison of different SSD techniques. Architecture of CNN is shown (refer Figure 6). Comparison of Deep learning-based CNN models are shown in Table II. The representation of YOLOV3 architecture is shown in (refer Figure 7). Table III shows a comparison of different YOLO techniques. Table IV represents OPENCV frameworks for face detection. Figure 8 demonstrates MTCNN architecture.

1) Single Shot Detector (SSD): SSD handles detections with two building blocks, a backbone, and head. Backbone extracts features from pre-trained ResNet model excluding fully convolutional layers and trained on benching dataset such as ImageNet. The SSD head comprises of convolutional layers following the backbone and the outputs consists of confinement boxes and objectness score in the spatial domain. To detect objects, the entire image is divided into grids. Anchor boxes enable detecting an object of a specific size and shape within the grid cell. Cricket ground maps to the vertically narrow anchor box while sky scrapper relates to horizontally thin one. The pre-defined cells are compared with ground truth for evaluation of intersection over union for estimating bounding box coordinates and object class. Zoom parameters determine the level to which the cell must be scaled to fit faces of diverse sizes for detection. A 4*4 grid, 2*2 grid and 1*1 grid detect smaller objects, mid-sized objects and the objects that cover the entire image respectively. Mediapipe Face detection supports multiple face detection along with key point detection, iris detection, face mesh detection and hair segmentation. It also supports pose estimation, object detection, and object tracking. Blaze face is a lightweight algorithm working effectively on mobile GPU enabled anchors.

SSD discretizes the image space into a set of default boxes with aspect ratios and scales [21]. At inference time, the system produces an objectness score for the existence of object in each default box and performs adjustments to better fit the object shape. Also, the system integrates estimations from multiple feature maps with varying resolutions to deal with objects of different sizes.

Feature Agglomeration Networks [22] are motivated by feature pyramid networks (FPN). The prime objective is to utilize multiscale features for aggregating higher level semantic feature maps to boost lower-level feature maps via hierarchical agglomeration.

To deal with multiscale faces, one way is to train multishot single scale detectors by using image pyramid to train multiple separate single scale detectors each for one specific scale. However, this is computationally expensive since it must pass through the network multiple times during testing. Another approach is to train a single shot multiscale detector by exploiting multiscale feature representations requiring only a single pass through the network while testing. Single shot scale invariant detector (S3FD) [23] contributes to face localization in the following three aspects 1. Scale equitable detection structure for multiscale face detection 2. Scale compensation anchor matching strategy for tiny face detection 3. Reducing false alarms with max out background label. It is common observation that anchor based detectors fail to detect tiny faces. Max out operation injects local optimal solution to deal with unbalanced binary classification problem of negative anchors (i.e. background) and only few positive anchors (i.e. face). This happens due to dense tile of small anchors, contributing to most of false positive faces.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 12 , December - 2025

TABLE I
COMPARISON OF SSD BASED FACE DETECTION METHODS. ALGORITHMS ARE EVALUATED IN TERMS OF FEATURE EXTRACTOR, DATASET, PERFORMANCE METRICS, LOSS FUNCTION, TRAINING DEVICE, AND FRAMES PER SECOND(FPS).

Sr No	Algorithm	Name	Feature Extractor	Dataset	Performance metric	Loss function	Training Device	FPS
1	REF[21]	SSD	Convolution filter	VOC2007	mAP-74.3%	Smooth L1 and Softmax	Nvdia TitanX	59
2	REF[22]	FANet	VGG-16	Pascal Face	AP-98.78%	Hierachical loss	Nvdia GPU GTX1080ti	35.6
3	REF[23]	S3FD	VGG-16	Pascal Face AFW FDDB WiderFace	AP-98.49% AP-99.85% TPR-0.983 AP-0.840	Multitask loss	Nvdia TitanX	36
4	REF[24]	SSH	VGG-16	Pascal Face FDDB WiderFace	AP-98.27% TPR-0.981 AP-0.8444	Multitask loss	Nvidia Quadro P6000	
5	REF[25]	DSFD	VGG-16	FDDB WiderFace	TPR-0.991 AP-0.90	Progressive anchor loss	Nvdia GPUP40	22
6	REF[26]	Pelee	Peleenet	Pascal VOC2007 MS-COCO	mAP-76.4% mAP-22.4%		Nvidia TX2	125
7	REF[27]	Pyramid Box	VGG-16	FDDB WiderFace	TPR-0.987 AP-0.887	Pyramid Box loss	–	–
8	REF[28]	RefineDet	VGG-16 Resnet-101	Pascal VOC2007	mAP-80.1%	Loss in anchor refinement module and object detection module	Nvidia TitanX	40.3
9	REF[29]	Fcos	Resnet-101 Resnet-50	MS-COCO	AP-44.7%	Focal loss and IoU loss	–	–
10	REF[30]	LFFD	–	FDDB WiderFace	TPR-0.973 AP-0.780	Softmax with cross entropy and L2 loss	Nvdia GPU GTX1080ti	136.99
11	REF[31]	SANet	Resnet-50	WiderFace UFFD	mAP-0.882 mAP-0770	Multitask loss	GPU	–
12	REF[32]	HAMBox	Resnet-50	Pascal Face AFW FDDB	AP-99.50% AP-99.9% AP-0.991	Focal loss	Nvdia Tesla V100	–

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 12 , December - 2025

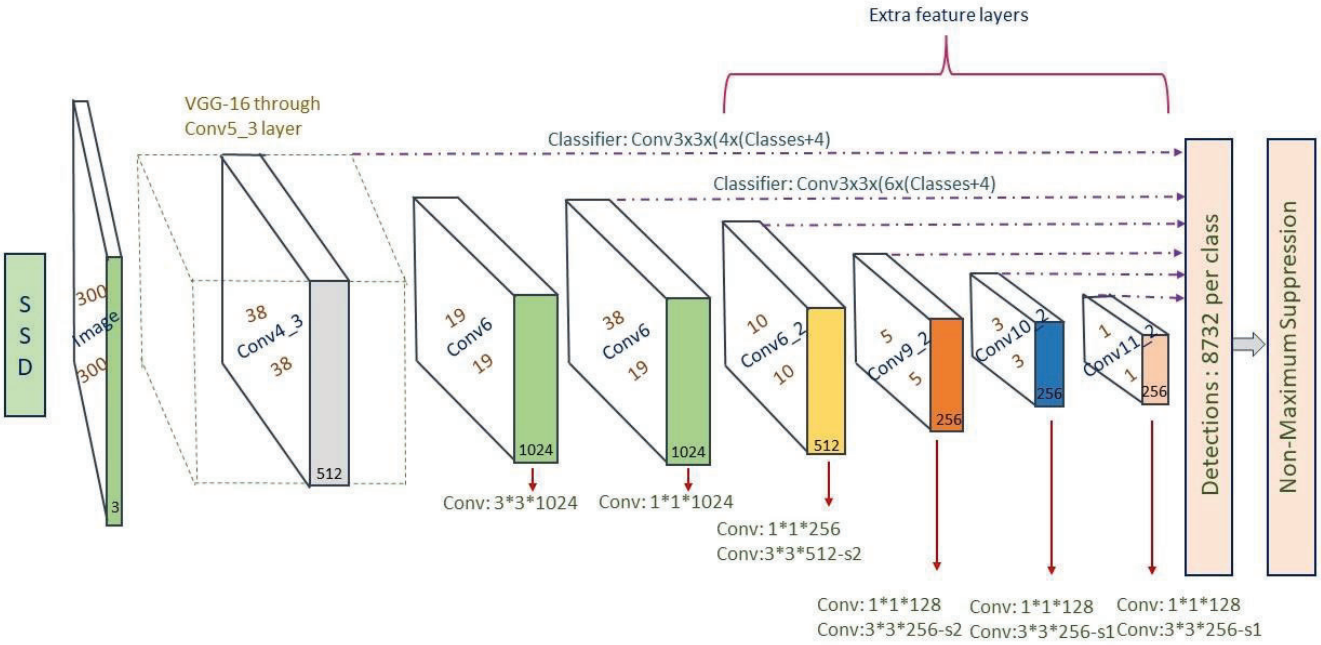

Fig. 5. The Architecture of SSD model

Single stage headless (SSH) [24] performs far better than state of art methods by removing fully connected layers (head) of VGG-16 backbone. SSH can detect faces at various scales without generating an image pyramid. SSH uses simple convolution to achieve larger window effect. During training phase anchors are assigned to three modules M1, M2 and M3 depending on face size. One unique aspect of this framework is that an anchor is assigned to ground truth face if and only if it has higher IoU than 0.5.

Dual shot face detector (DSFD) [25] is a variant of single stage detector comprising of two stream design networks. It is an efficient face detection framework consisting of complimentary modules such as feature enhancement module (FEM), progressive anchor loss (PAL) and improved anchor matching (IAM). FEM ensures discriminability and robustness of features, PAL combines hierarchical loss and pyramid anchor that assigns smaller anchor sizes in first slot and larger sizes in second shot, IAM uses anchor partition strategy and anchor-based data augmentation to better match anchors and ground truth faces for effective regressor initialization.

Pyramid Box [27] address the challenge of unconstrained face detection. To deal with hard face localization contextual data is exploited with context anchor, low level feature pyramid network and context sensitive structure. Multiscale training samples are generated from data anchor sampling to ensure diversity of data for tiny faces. Devised Receptive Field Block net (RFB-net) [33] motivated by the receptive fields (RF) in human beings to extract discriminable and robust features. Pelee [26] is another efficient single stage detector that handles tradeoff between computer power memory resources [28] inherits one stage approach consisting of anchors and object detection network to adjust size of anchors. Features from anchor refinement module are forwarded to object detection network via a transfer connection block. Blaze face [29] is a novel light weight face detector featuring super embedded devices of 200-1000+ fps on flagship embedded devices based on mobile level v1/v2. Fully convolutional one stage object detection (Fcos) [30] performs pixel object detection analogous to sematic segmentation. Fcos is anchor box free as well as proposal free. Fcos meticulously avoids hyper parameters and complex computation concerned with anchor boxes. Anchor based detection is sensitive to scale, aspect ratio and location of anchor boxes. Low quality anchor boxes are discarded by evaluation of centerness of a location. At inference time, the classification scores get down-valued when multiplied by center-ness to reject low quality predicted bounding boxes. Light and fast face detector (LFFD) [31] enables deployment on edge devices utilizing receptive field and extended receptive field. Anchor based unable to cover all face scale, threshold for IOU is empirically set, sample imbalance and redundant computation.

For large faces (RF), for medium, faces (ERF with little context) for small faces (ERF with relevant context). Pyramid Box ++ [32] is extension work based on pyramid box boosting face detection with balanced data anchor sampling, dual pyramid anchors and dense context module.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 12 , December - 2025

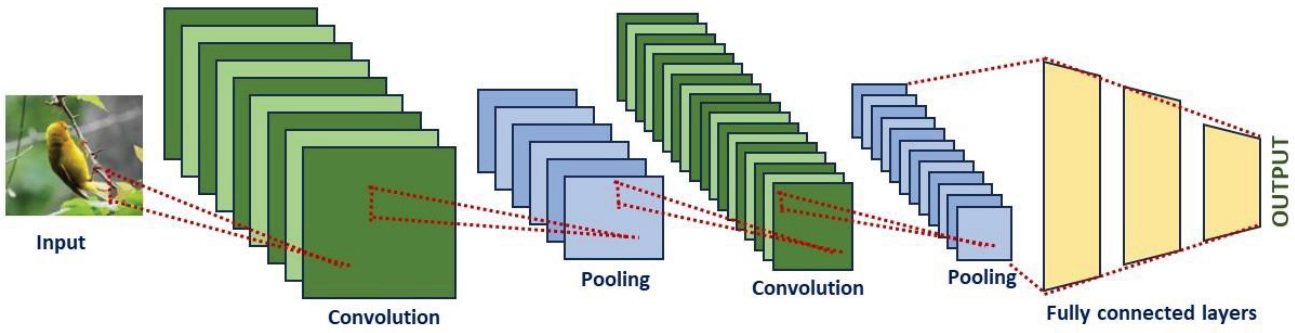

Fig. 6. Architecture of Convolutional Neural Network (CNN)

Integration of multiscale features can contribute significant noise. In smooth attention network (SANet) [34] attention guided feature fusion module [AFFM] and smoothed context enhancement module (SCEM) manages the griding artifacts. AFFM performs attention wise feature fusion artifacts. AFFM performs attention wise feature fusion of high and low level to reduce noise. SCEM manages the griding artifacts contributed by deconvolutional layer to preserve local spatial information.

2) Convolutional Neural Networks (CNN): Convolutional neural networks are the most successful and commonly used algorithms in computer vision tasks. CNN basically consists of three parts convolution layers: it consists of kernels of varying sizes and are used for feature extraction, non-linear layers consist of an activation function that deals with the nonlinear functions, pooling layer to get statistical information about the neighboring pixels. The convolution layer typically consists of kernels (convolutional filters) sliding across the dimensions of an image. Feature extraction is a mathematical operation performing dot product between corresponding values of the image and the convolutional filter. The main advantage in using CNN is that the receptive fields share the same kernels, minimizing the memory constraints as compared to deep neural networks. The main task of the pooling layer is sub-sampling and gathering local statistical information from the feature maps. Equivalent representation, sparse representation and parameter sharing are three key benefits of CNN.

[35] is the first to show that CNN can yield better performance on Pascal VOC as compared to the system based on HOG. The proposed system has three subsections comprising of category independent region proposals utilizing selective search, large CNN for fixed length feature extraction and set of SVM's. During inferencing 2000 category independent region proposals are generated from input image. [36] pointed out the trade-off between strong discriminative features and efficient computation. To promptly reject background regions and improve localization capabilities, a calibration stage is introduced after detection stage. [37] [38] showcased that regional proposals are computationally intensive.

RPN (Region proposal network) is a fully convolution network estimating object boundary and objectness confidence simultaneously thereby enabling faster inferencing. [39] [40] introduced attribute aware face detection, Faceness-Net. Partness map relates the presence of specific facial component in the image. [41] developed contextual multiscale R-CNN (CMSRCNN) by additionally providing context information to identify difference between real faces with bodies and fake face without bodies. Here multiscale is rooted both in region proposal and ROI layer to deal with tiny faces. Also, contextual reasoning is added for challenging face detection.

Lower layer features correspond to edges and corners pertaining to localization information. Deeper layer features are class – dependent contributing to high end tasks of face detection. [42] demonstrated the role intermediate layers features called hyper features for training of different tasks under consideration. Features common to tasks can be combined with feature fusion technique translating the features to a common

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 12 , December - 2025

subspace by linear or non- linear combination.

A Supervised Transformer Network [43] employing a cascaded CNN for RPN to predict face regions and corresponding landmarks. Facial landmarks are warped with candidate regions to obtain canonical positions of face. Finally, RCNN verifies the candidate regions for valid or invalid face.

Detection accuracy is improved with the development of several techniques including position sensitive average pooling, Multiscale training and testing and on-line hard example mining strategy in R-FCN [44]. R-FCN improved detection by introducing additional small anchors and modified the position sensitive ROI pooling to a smaller size for tiny face detection. Position sensitive average pooling was used instead of normal average pooling for last layer for enhance embedding. In R-FCN the feature maps are more expressive as unnaturally injecting fully connected layers into Resnet is avoided and easier learning of class score and bounding box is accomplished.

Conventional CNN simply consists of stack of filter layers where input passes through all of them before reaching classifier. It is the well-known fact that deeper layer possess discriminative capabilities and lower layer rejects non-face samples.

[45] proposes Inside cascaded structure that introduces face/ non-face classifiers at different layers within same CNN. To determine which samples should be passed to the data routing layer. Early rejection classifier (ERC) predicts face probability to determine which samples should be passed to data routing layer. Data Routing is a mechanism where different layers are trained via different samples, deeper layers dealing with more difficult Samples. Along with this, contextual CNN boosts the detection accuracy by usage of body part information.

In [46] presents Scaleface network that does not require image pyramid (IP) having moderate complexity. The design of appropriate receptive fields is essential for multiscale face detection. Recent methods can be categorized into two classes: Scale Variant based method and Scale Invariant based methods. The [47] proposes an improved faster RCNN framework by combining various techniques including features concatenation, hard negative mining, Multiscale training, model pretraining and proper calibration of key parameters. A novel approach in [48] derives robust face detection methodology face RCNN based on faster RCNN. One key technique considered here is bootstrapping with online hard example mining OHEM. The key idea is to collect hard samples and feed them again to the network to strengthen discriminative power. The loss function represents how effectively the network performs, the generated proposals are sorted by their losses and top N worst performing examples are taken as hard examples.

Face Boxes is a powerful lightweight network structure comprising of Rapidly Digested convolutional layers (RDCL) and multiscale convolutional layers (MSCL)[49]. RDCL enables real time face detection and MSCL handles faces over various scales. RDCL is designed to quickly reduce input spatial size by suitable kernel. The anchors of RPN are associated with last convolutional layers whose features and resolution are too weak for handling faces of diverse sizes. Also, anchor associated layer detects face within corresponding range of scales and has single receptive field that cannot cope up with different scales. MSCL takes care of problems related to RPN by discretizing anchors over multiple layers with different resolution to handle faces of various sizes.

DSFD (Different Scale Face Detector) [50] technique handles small, and scaled faces. Feature maps for small faces shrink gradually over the depth of convolutional neural network and can hardly detect faces less than 15x15 pixel.

3) You Only Look Once (YOLO): Conventional classifiers are modified and remodeled to perform localization. The algorithm applies the model at multiple locations and dimensions. Candidates with high scores are treated as detections. Yolo Face is variant derived from Yolov3.Yolo performs single stage detections by dividing the image into grids and estimating the bounding boxes with predicted probabilities. The model glances at the entire image during test time, so predictions are guided by global context. Yolo divides the image into grids with centerpoints. The number of grid sizes range from 3*3, 4*4 and 16*16, there is no fixed rule for number of grids. Each grid is looked for face detection only at one time in a forward pass. Therefore, it is called You Look Only Once. For the summary of YOLO development stages (refer Figure 9).

Non maximum suppression is entrenched on evaluating the maximum overlap ensuing a unique bounding box. Yolo utilizes CNN layers with stride two as against pooling layers capturing low level features to detect small objects. The earlier approach for object detection is based on classifiers and objectness score. In YOLO a single stage framework estimates the bounding boxes and class probabilities for an object under consideration. Here, object detection is formulated as a regression problem unlike classification approach considered earlier. The development of YOLO series is analyzed across depth and breadth in [51] [52]. The YOLO versions for face detection are designed in [53] [54] [55]. YOLO can be efficiently implemented on FPGA benefiting from its computation capabilities [56] [57]. YOLO-LITE [58] enables the deployment of YOLO algorithm on portable devices.

4) Multitask Cascaded Convolutional Neural Network (MTCNN): MTCNN consists of three CNN assigned a distinct role and successor refines the detections. Stage-I: P-net scales the picture to enable detections for diverse face range. Convolution operation with 12*12 kernel progressively traverses through all scaled pictures detecting for faces and respective locations. Multiple redundant detections can be mitigated with the application of non-maximum suppression. NMS criteria may be a large bounding box or large confidence. Stage-II: R-net If the bounding box is out of bounds, the portion of image inside bounding box is copied to new array and remaining everything is filled with a zero. R-net handles bounding box extending beyond image boundaries by creating a new array and setting out of bound values to zero. R-net confines the exact face detection bounding boxes eliminating the redundant ones and returning appropriate square shape boxes. R-net output is like P-net, but it includes a new, more accurate one.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 12 , December - 2025

TABLE II
COMPARISON OF CNN BASED FACE DETECTION METHODS. ALGORITHMS ARE EVALUATED IN TERMS OF FEATURE EXTRACTOR, DATASET, PERFORMANCE METRICS, LOSS FUNCTION, AND TRAINING DEVICE.

Sr No	Algorithm	Name	Feature Extractor	Dataset	Performance metric	Loss function	Training Device
Sr No	Algorithm	Name	Feature Extractor	Dataset	Performance metric	Loss function	Training Device
1	REF[59]	TinaFace	Resnet-50	WiderFace	AP-92.4	Focal loss, DIoU loss, crossentropy loss	NVIDIA GeForce GTX 1080 Ti
2	REF[60]	RetinaFace	Resnet-50	WiderFace	AP-91.7	Smooth L1loss	NVIDIA Tesla P40
3	REF[61]	Mask RCNN	Resnet-101	FDDB AFW WiderFace	TPR-0.880 AP-95.97 AP-0.662	Multitask loss	NVIDIA GeForce GTX 1080 Ti
4	REF[62]	RefineFace	Resnet-50	WiderFace AFW Pascal Face FDDB	Accuracy-91.1 AP-99.90 AP-95.45 TPR-0.9911	Scale aware margin loss	NVIDIA GTX 1080 Ti
5	REF[50]	DSFD	VGG-16	MAFA FDDB WiderFace	Accuracy-95.7 Recall rate-99.22 Recall rate-98.12	Least square	NVIDIA GTX Titan-X
6	REF[49]	FaceBoxes	Inception3	AFW PASCAL FDDB	AP-98.91 AP-96.30 TPR-0.960	Softmax loss, Smooth L1 loss	–
7	REF[38]	Faster RCNN	Resnet-101	WiderFace	AP-87.9	Multitask loss	NVIDIA Tesla K80
8	REF[48]	Face RCNN	ConvNet, VGG-19	WiderFace	AP-0.827	Center loss, multitask loss	–
9	REF[44]	Face R-FCN	Resnet-101	FDDB WiderFace	TPR-0.9907 AP-0.876	Softmax loss, Smooth L1 loss	–
10	REF[42]	Hpyerface	Resnet-101	FDDB AFW Pascal Face	TPR-0.901 AP-97.9 AP-92.46	Eucledian loss, Softmax loss	–
11	REF[41]	CMS RCNN	VGG-16	FDDB WiderFace	TPR-0.906 AP-0.643	Multitask loss	–
12	REF[46]	ScaleFace	Resnet-50	FDDB WiderFace	Recall rate-96% AP-76.4 %	–	NVIDIA Titan-X

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 12 , December - 2025

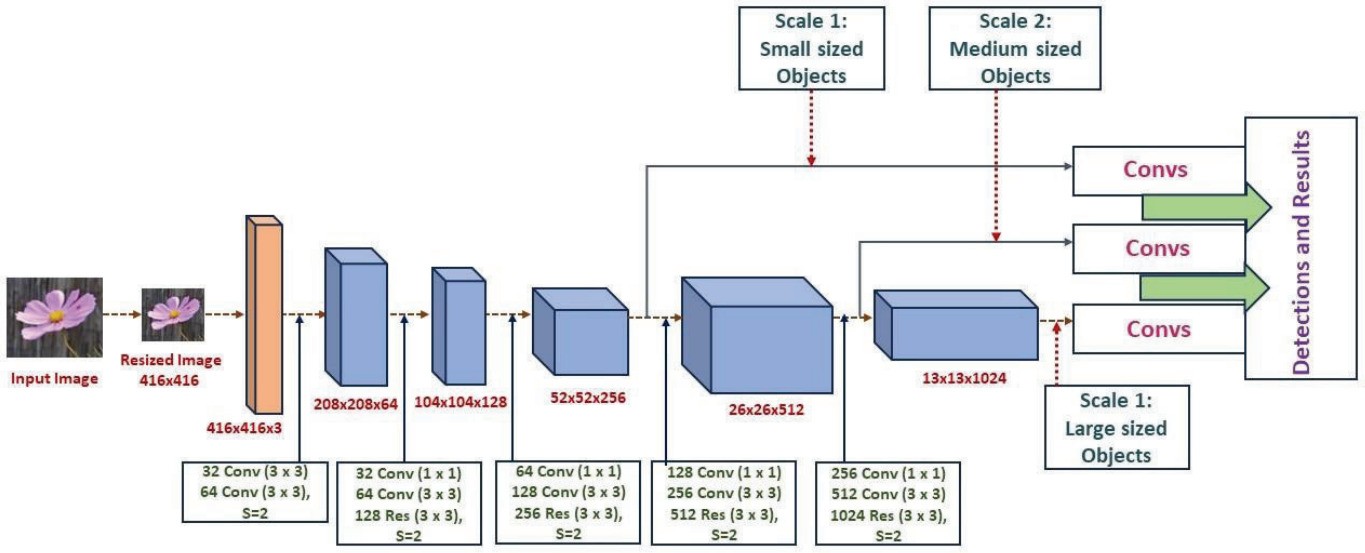

Fig. 7. Architecture of You Only Look Once (YOLO)

5) OpenCV Face detection methods: In [69] detections are performed merging OpenCV library with CAFFE and TensorFlow framework. OpenCV performs the face detections with two options.

Before deployment of an application on embedded platform, the critical factors affecting the performance of the system must be considered. We have presented exhaustive discussion on several deep learning approaches for face detection. The success of any application depends largely on utilizing the advantages and eliminating the disadvantages with software optimization. Table V shows the benefits and drawbacks of various deep learning methods. Also, there exists methods that are not a part of the categories considered above or typical methods for handling constraints related to face detection. Multitask face detection [70] [71] [72], Tiny face detection [73] [74], face detection analysis toolkit [75] [76], Occlusion efficient face detection [48] [77] and rotation invariant face detection [78] are popular methods for role specific face localization.

III. Overview Of Different PYPI Software Package

Face Detection implementation is reaching new heights with the usage of different python packages. The packages can be installed on computing devices and algorithms can be executed with a simple command line interface.Different software packages for face detection are analyzed in Table VI. Figure 10 depicts different face detection software implementation available. Discussion of detailed architecture of different models and their variants gave a deep insight for underlying principle behind the detection mechanism. Before selecting a particular model for an application, one must be aware of the limitations and benefits each one has, as provided in Table VII. Inaccurate face detection affects the performance of an application. Figure 11 contributes for factors resulting in inefficient face detection based on underlying architecture and performance metrics. The success of the practical computer vision applications is dominated by haar cascade classifiers due to their fast execution time. Several algorithms based on hand crafted features utilizing machine learning ideology emerged for effective face detection. In 2012, a deep learning era emerged contributing to the most accurate and quick face localization techniques. It is crucial to interpret detailed architecture of different models and their variants to get a deep insight into the underlying principle. Custom face detection is essential to overcome the specific challenges related to tiny faces, blurry faces, modality, illuminated faces, rotated faces, extreme expression faces. Custom Face Detection can be implemented with Yolov5, Yolov7, TensorFlow and Detectron models, to surpass the limitations mentioned beforehand. Face Detection can be implemented with either extraction of hand-crafted features or utilizing some deep learning based pre-trained models. Custom face detection models can withstand diverse real time conditions encountered in day-to-day life. To start with, one can go for pre-trained models. If the desired accuracy is not achieved then one can test for custom models.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 12 , December - 2025

TABLE III
COMPARISON OF YOLO BASED FACE DETECTION METHODS. ALGORITHMS ARE EVALUATED IN TERMS OF FEATURE EXTRACTOR, DATASET, PERFORMANCE METRICS, LOSS FUNCTION, AND TRAINING DEVICE.

Sr No	Reference	Variant	Backbone	Neck	Head	Loss	Improvements
1	REF[63]	YOLOv1	GoogleNet	–	FC–>7x7x(5+5+50)	MSE	Directly fit the location of the bounding box
2	REF[64]	YOLOv2/ YOLO9000	Darknet-19	–	Conv–>13x13x5x(5+20)	MSE	Batch Normalization, High Resolution classifier, Convolutional with anchors, Dimension clusters, Direct location prediction, Fine-Grained features, Multiscale training, Hierarchical classification
3	REF[65]	YOLOv3	Darknet-53	FPN	Conv–>13x13x5x(5+80) –>13x13x5x(5+80) –>13x13x5x(5+80)	MSE	Multiscale prediction, Better classification network, Binary cross entropy loss
4	REF[66]	YOLOv4	CSPDarknet-53	SPP+PAN	Conv–>13x13x5x(5+80) –>13x13x5x(5+80) –>13x13x5x(5+80)	CIoU	Mosiac for data enhancement, Using multi anchors for single groundtruth, Eliminate grid sensitivity (sigmoid), MSE loss–>GIoU loss–>CIoU loss
5	REF[63]	YOLOv5	Focus CSP Darknet-53	SPP+PAN	Conv–>13x13x5x(5+80) –>13x13x5x(5+80) –>13x13x5x(5+80)	GIoU	Adaptive anchor strategy, Adopt Focus Structure, CSP Structure
6	REF[67]	YOLOv6	RespBlock CSPStackRep Block	PAN	Decoupled head	GIoU SIoU CIoU	More training epochs, Self Distillation, Gray border of images
7	REF[68]	YOLOv7	VOVNET CSPVOVNET ELAN E-LAN	PAN	Lead head, Auxillary head	IoU	Bag of freebies with batch normalization, Implicit knowledge, EMA model

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 12 , December - 2025

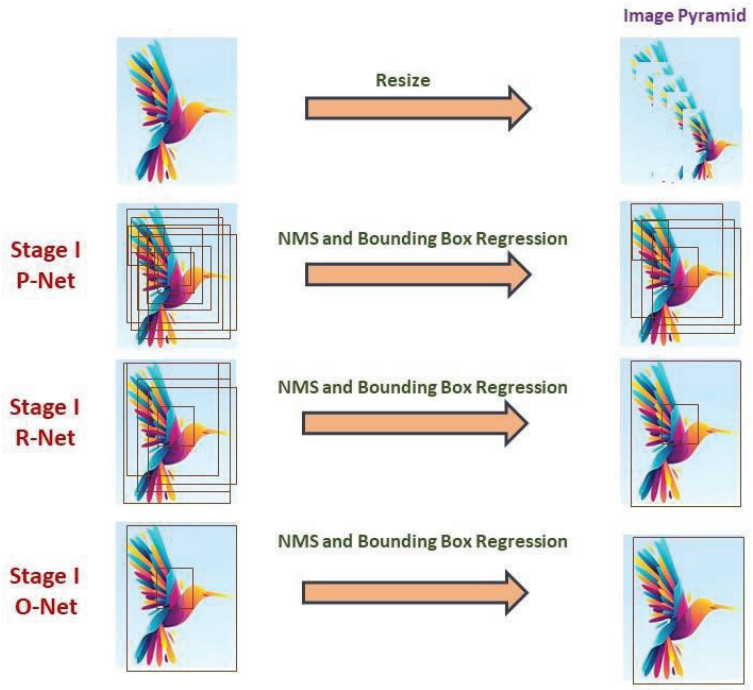

Fig. 8. Architecture of MTCNN model

TABLE IV
OPENCV FACE DETECTION METHODS

Sr. No	Framework	Memory	Function
1	Caffe	5.10 MB	cv2.dnn.read NetFromCaffe()
2	TensorFlow	2.7 MB	cv2.dnn.readNet FromTensorflow()

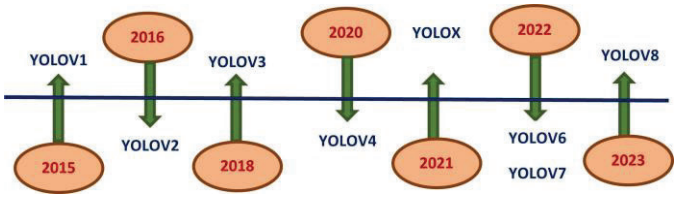

Fig. 9. Evolution of YOLO variants

IV. OVERVIEW OF INTELLIGENT COMPUTING TECHNIQUES

The information generated from end terminals such as cameras needs to be processed for analysis or forwarded to train a model. Expeditious training and inferencing demand significant computation resources. Training a deep learning model is a complex process involving millions of parameters and high dimension data. Cloud computing leverages the critical computational resource constraint by relocating data to centralized cloud [79]. Cloud computing is a phenomenon of utilizing remotely located servers on the internet to handle end user data. Cloud computing demands low latency, scalability, and privacy [80]. Table VIII presents different cloud instances with support for computer vision application development. For Fast Tag applications camera frames need to analyze instantly to initiate a payment process. Handling this application through cloud will incur overhead in terms of processing and forwarding delay. Addition of a greater number of users can lead to bottleneck for bandwidth sensitive applications. Sending personal data to the cloud raises safety concerns and can violate privacy aspects.

Edge computing overcame the above challenges by implementing the computing nodes close to the source. Edge computing facilitates processing and management of client data at periphery of network in the proximity of the originating source. Edge computing demands a high-end computer node, heterogenous communication and privacy. Google released TensorFlow and TensorFlow Lite framework for implementation on edge devices. Facebook developed Caffe and Caffe2 framework for deployment of deep learning models on Raspberry Pi and mobile devices. PyTorch is another framework by Facebook to promote production of research prototypes. NVIDIA GPU powered by CUDA and cuDNN promotes parallel computing ability on self-developed embedded platforms such as NVIDIA Jetson Nano, NVIDIA Xavier, and NVIDIA Orin. To integrate the model on embedded devices, models with reduced parameters can boost memory and latency

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 12 , December - 2025

TABLE V

SUMMARY OF BENEFITS AND DRAWBACKS OF DEEP LEARNING BASED FACE DETECTION METHODS. BENEFITS ENABLES THE END USER TO CHOOSE AN APPROPRIATE ALGORITHM DEPENDING ON THE RESOURCES AVAILABLE. DRAWBACK PUTS CHECK ON THE COMPUTING DEVICES PROCESSING CAPABILITY

Sr. No.	Reference	Method	Benefit	Drawback
1	REF[21]	SSD	Single stage computation eliminating proposal generation and feature resampling stages. Faster than YOLO. Faster predictions at different scales using multiple layers achieving high accuracy even at low resolution input.	Multiscale feature maps alone are used for prediction and thus high resolution semantically weak feature map may fail to perform accurately.
2	REF[38]	Faster RCNN	Accurate and Robust due to explicit region proposal and pooling.	Increased computation time as proposals is generated in the first stage and classified in second stage. The fastest detector operates at only 10's of fps. ROI pool layer builds features from last layer and unable to detect small faces.
3	REF[22]	FPN	Merges low resolution semantically strong feature with high resolution semantically weak feature	Ignores context information between anchors without monitoring current layers information for feature fusion
4	REF[41]	CMS-RCNN	Better feature extraction with larger window by duplication of head. Boosts detection accuracy by using contextual body part information.	Increased memory requirement and detection time
5	REF[24]	SSH	Simple convolution to achieve larger window effect. The anchor is assigned a ground truth face if and only if it has IoU higher than 0.5. Face detection at various scales without generating image pyramid.	Inefficient tiny face detection.
6	REF[30]	FCOS	Anchor box free and proposal free. Avoids some hyperparameters and complex computation concerned with anchor boxes.	Feature maps for small faces gradually shrink over the depth of the network and can hardly detect faces of smaller dimension.
7	REF[35]	R-CNN	Enhanced face localization due to region proposals generated by Selective search	Region proposal is bottleneck
8	REF[40]	RPN	Estimates object boundary and objectness score simultaneously resulting in faster inferencing.	The anchors are associated with last convolutional layer whose feature resolution is too weak for handling faces of diverse size. Anchor associated layer detects faces within corresponding range of scales and has single receptive field that cannot cope up with different scales.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 12 , December - 2025

TABLE VI
PHYTON PACKAGES FOR FACE DETECTION. DETAILS ABOUT METHOD, WEIGHT FILES, CONFIGURATION FILES, FACE DETECTION FUNCTION, PARAMETERS TO THE FUNCTION AND OUTPUT OF EACH METHOD IS PRESENTED. COVERS BOTH MACHINE LEARNING AND DEEP LEARNING METHODS.

Sr No	Method for Face Detection	Technique Used	Pre-trained model/ Configuration / Weight file/Package	Face Detection function	Parameters to function	Output
1	OpenCV Haar Cascade Face Detection	Haar features	Xml file, 941 KB	haar_cascade. detectMultiScale()	Image, scaleFactor, minNeighbors	Bounding Box co-ordinates
2	Dlib HoG Face Detection	Histogram of oriented gradients(HoG)	HOG + Linear SVM face detector using dlib 18.51 MB	dlib.get_frontal _face_detector()	Image, upsample	Bounding Box co-ordinates
3	OpenCV Deep Learning-based Face Detection (CAFFE Framework)	ResNet	deploy.prototxt 28 KB, res10_300x300_ ssd_iter_140000. caffemodel 10417 KB	opencv_dnn_ model.forward()	Preprocessed image, prototxt-model configuration, caffemodel-model weights	Normalized Bounding Box co-ordinates
4	OpenCV Deep Learning-based Face Detection (TensorFlow Framework)	OpenCV	opencv_face_ detector_uint8.pb 2664 KB, opencv_ face_detector.pbtxt 35KB	opencv_dnn_ model.forward()	Preprocessed image, pbtxt-model configuration, pb-model weights	Normalized Bounding Box co-ordinates
5	Dlib Deep Learning based Face Detection	Histogram of oriented gradients (HoG)	mmod_human_ face_detector.dat 713 KB	dlib.cnn_face_ detection_ model_v1()	Model_path	Bounding Box co-ordinates and confidence
6	Mediapipe Deep Learning-based Face Detection	Single Stage Detection: Single Shot Multibox Detector(SSD	mediapipe/graphs /face_detection/ face_detection_ desktop_live.pbtxt 49.8 MB	mp_face_ detection.Face Detection()	Image, model_selection, min_detection_ confidence	Normalized Bounding Box co-ordinates, normalized keypoints and confidence
7	YoloFace Detection	You Only Look Once(YOLO)	yolov3_tiny face.cfg and yolov3 tiny_face .weights 237 MB	face_analysis .face_detection()	Image, model	Bounding Box co-ordinates and confidence
8	MTCNN based Face Detection	Convolutional Neural Networks (CNN)	mtcnn package 3.05 MB	detector.detect _faces()	Image	Bounding Box co-ordinates, keypoints and confidence
9	RetinaFace Face Detection	Single Stage Detection	retina-face package 85.831 KB	RetinaFace .detect_faces()	Image	Score, Facial area and Landmarks
10	DSFD and Retinaface Face Detection	Single Stage Detection	face-detection package 77.731 KB	face_detection. build_detector()	DSFDDetector, confidence_threshold, nms_iou_threshold	Bounding Box co-ordinates and detection confidence
11	yolo5face Detection	You Only Look Once(YOLO)	yolo5face package 53.868 KB	get_model()	Model, gpu, target_size, min_face, image	Boxes and keypoints

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 12 , December - 2025

TABLE VII
SELECTION OF AN ALGORITHM FOR AN APPLICATION IS CLOSELY RELATED TO BENEFITS AND LIMITATIONS IT OFFERS. SUMMARY RELATED TO DIFFERENT PRACTICAL METHODS COVERING MACHINE LEARNING AND DEEP LEARNING-BASED APPROACHES IS PRESENTED.

Method	Benefits	Limitations
Haar Cascade or Viola Jones	Faster detections on real time CPU	Does not work on non-frontal and occluded faces. Tradeoff between false positives and 'minNeigbours' argument. Tradeoff between inference time and 'scaleFactor' parameter.
HoG with SVM	Works on slightly rotated faces and small occlusions.	Fails on profile faces and occluded faces, estimating bounding boxes debarring facial parts. Inference is worsened further if the input image is up sampled.
OpenCV	Highly accurate than the above two methods. Higher speed attained with multiscale detection permitting resizing and optimal computations.	It cannot detect faces with smaller dimension.
CNN	Accurate and robust than conventional methods. Detections under varying face orientations and lighting.	High detection time limits usage for real world applications. Inference is worsened further if the input image is scaled using image pyramid.
YoloFace	Faster than conventional CNN.	Fails for tiny faces
MTCNN	Faster as compared to conventional convolution neural network approach using DLIB library.	Detections are affected by extreme lightning and massive occlusions.
SSD	Super real time performance, this makes it unique from other methods.	Inaccurate key point regression for occluded faces.

constraints. YOLO, MobileNet, SSD and SqueezeNet enable device computation with simple mathematics and single stage operation. Model compression with pruning, knowledge distillation and quantization shrink the model at the cost of reduced accuracy. An alternative to GPU is FPGA [81], allowing reconfiguration leading to customized architectures and enabling model level optimization. Google's Tensor processing unit (TPU) is leveraging deep learning inference. Programming of FPGA and ASIC enforces knowledge of hardware level abstraction. OPENCL empowers software level programming of FPGA in contrast to hardware level programming.

The potential of intelligent computing platforms for data preprocessing, model training, model deployment, parallel computing, and brisk inferencing can be harnessed in the automotive domain for a customized task. State of the art research in Computer vision for ADAS is gaining popularity and has a promising future with the integration of cameras for monitoring. The overall driver monitoring system revolves around the interpretation and processing of the video frames collected by the camera for emotion detection, drowsiness detection and driver authentication. Face detection is the first step towards the development of driver monitoring systems and advanced driver assistance systems. Monitoring and surveillance systems require identification of passengers in the vehicle. The acquisition of frames from multiple streams demands swift training and inference. Table IX projects machine learning based ADAS systems. Figure 12 and 13 shows computing platforms for ADAS.

V. CHALLENGES AND SCOPE FOR RESEARCH

A. Unconstrained Face Detection

Unconstrained face recognition has advanced quickly, reaching saturation in recognition accuracy for the benchmark datasets used today. While crucial for early development, the use of a generic face detector for face imagery is a major drawback in most benchmark datasets. Restricted variations in face posture and other confounding elements are the implication of this method. Classical face detectors toil with constraints of limited variation in pose, occlusion, expression, reflection, styling accessories. Tremendous scope towards development of exhaustive unconstrained face detection dataset and effective models is the need of an hour.

B. Deployment on Embedded platforms

Training of face detection model and inferencing demands fast processing on CPU or GPU or ASIC or FPGA. Some of these devices support parallel computing to enhance detection performance. Some edge specific tools are supported exclusively for their own computing devices. NVIDIA GPU leverages model deployment on NVIDIA specific platforms only such as NIVIDIA Jetson, NVDIA Orin, and NVIDIA Xavier. Similarly, Intel OPENVINO toolkit supports Intel chips only. Compression of the deep learning model facilitates usage on edge nodes. Parameter quantization, pruning, knowledge distillation, and fast exiting are popular methods for reducing the model. Although tremendous progress is ongoing for model deployment and compression, still wide scope exists for further improvement. Two avenues are open for research to

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 12 , December - 2025

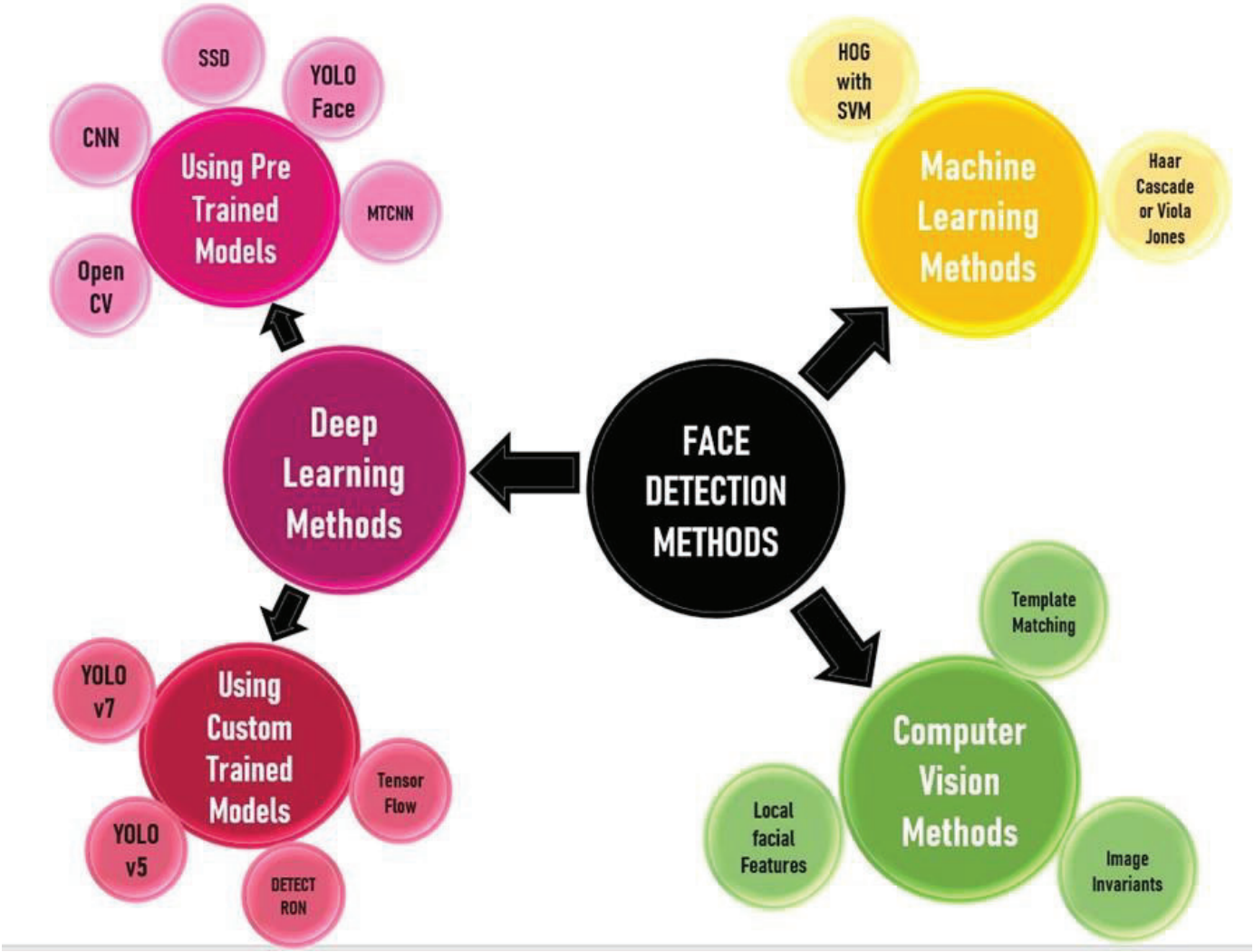

Fig. 10. Face Detection methods

allow heterogenous cross platform implementation on different embedded devices and compressing the model for minimal resource requirement as well as to boost accuracy and real time performance.

C. Hardware and Software Acceleration

To enhance the inference time, hardware suppliers are utilizing CPU, GPU, TPU, ASIC, and FPGA. CPU and GPU are flexible for diverse data handling but demands higher power consumption. In contrast, FPGA and ASIC supports reconfigurable design with lesser power requirements. FPGA programming requires hardware level knowledge and most of the researcher fail to have these fine details. OPENCL has enabled software level development of FPGA. Researchers are attracted towards software specific development due to user friendly tools, adaptable and configurable models, and command line execution interface. FPGA development can successfully enable researchers to explore for architectural variations. Data scientists can collaboratively work on hardware and software acceleration to enable configurable model design and easy implementation.

D. Latency, Energy and Migration on Edge platforms

Accommodation of data from many sources on cloud puts limitations on latency, energy, and migration. DNN (Deep Neural Network) partitioning performs effectively on sequential design but execution for recurrent neural network (RNN) is substandard. Optimal energy consumption is key aspect for hardware platforms (GPU, TPU) and its interaction with battery management systems must be dealt with to lessen power consumption. Virtual machine and Docker containers are the solutions for migration on edge servers, handling migration of deep learning models is still an area to explore. What module of DNN model must be migrated, what modules must be retained on virtual image, and can the modules be migrated during training? Dealing with the migration, system monitoring and analysis is crucial to gain empirical knowledge of the challenges.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 12 , December - 2025

TABLE VIII
SUMMARY FOR CLOUD INSTANCES WITH AVAILABLE FRAMEWORK, BENEFITS, KEY FEATURES, CUSTOMERS, PRICING, OBJECT STORAGE, SUPPORT FOR COMPUTER VISION APPLICATIONS.

Sr No	Cloud Instance	Details	Framework	Key Features/ Benefits	GUI Development Tool	Customers	Pricing	Object Storage	Vision Tasks
1	AWS Sagemaker	Amazon SageMaker provides image processing algorithms for image classification, object detection, and computer vision.	Apache MXNet, Apache Spark, Chainer, Hugging Face, PyTorch, scikit-learn, SparkML Serving, TensorFlow, Triton Inference Server.	Augmented AI, Autopilot, Clarify, Batch Transform, Data Wrangler, Edge Manager, Feature Store, Ground Truth, Elastic Inference, Model Monitor, Model Registry, Serverless Endpoints	SageMaker Canvas	Acquia, Allergan, Reamp, Nanotronics, Zendesk, Lyft, Redforce, Salesforce, Volkswagen, Netflix, Snap Inc.	Yes	Amazon S3	Yes
2	Azure Machine Learning	Azure Machine Learning empowers to build, deploy, and manage high-quality models faster and with confidence.	PyTorch, TensorFlow, scikit-learn	Synapse Analytics, Cognitive Services, Speech, Language, Computer Vision, Custom Vision, Language Understanding, Form Recognizer, App-Service-UI, NVDIA GPU-Optimized VMI.	Drag-and-drop machine learning	AXA, FedEx, 3M, NHS, Pepciso, Seven Bank	Yes	Blob Storage	Yes
3	IBM Softlayer (New name: IBM Cloud)	IBM Watson® Studio empowers to build, run and manage AI models, and optimize decisions anywhere on IBM Cloud.	PyTorch, TensorFlow scikit-learn	Classic Watson Assistant, Discovery v2, Knowledge Studio, Language Translator, Natural Language Understanding, Text to Speech, Speech to Text, Watson Assistant for IBM Cloud Pak for Data (Installed)	Drag-and-drop AI models	Allianz, AAIS, Assima, contextor, Deloitte, FaceMe, fitzsoft	Yes	IBM Cloud Storage	Yes
4	Google Cloud AutoML	Build, deploy, and scale more effective AI models with our Vertex AI.	TensorFlow, PyTorch, scikit-learn, along with supporting all ML frameworks via custom containers.	AutoML, Deep Learning VM Images, Workbench, Matching Engine, Data Labeling, Deep Learning Containers, Explainable AI, Model Monitoring, Neural Architecture Search, Pipelines, Prediction, Tensorboard	–	PayPal, P&G, King, Blue Apron, EQUIFAX, Sky, Magalu, ups, Etsy, Carrefour, gojek.	Yes	Google Cloud Storage	Yes

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 12 , December - 2025

TABLE IX
MACHINE LEARNING ADAS SYSTEM WITH DETAILS ABOUT DEVELOPER, SUPPORTED FRAMEWORK, INFERENCE ENGINE, DEPLOYMENT PLATFORMS, KEY APPLICATIONS AND FEATURES, SUPPORT FOR COMPUTER VISION TASKS.

Sr no	About the system	Developer	Frameworks	Inference Engine	Deployment/ Development platform	Key Applications	Additional Applications/ Features
1	High-Performance Automotive Vision Processing and Streamlined Development Efficiency for Advanced Driver Assistance Systems (ADAS). Accelerating innovation in automotive vision technology is fueling a transformation ADAS and will ultimately help to enable the achievement of fully autonomous L5 vehicles.	NXP	Pytorch	DeepViewRT, TensorFlow Lite, TensorFlow Lite Micro, Glow and ONNX Runtime	NXP Edge Verse i.MX RT crossover MCUs, and i.MX family application processors	Front View Camera, Surround View, Driver monitoring system, Driver Occupant system	Pedestrian detection, Lane keeping/ departure warning and assistance, Traffic sign recognition (TSR), Collision warning and avoidance, Blind spot monitoring
2	The NVIDIA DRIVE platform is a full-stack solution that spans fully autonomous, highly automated, and supervised driving. It includes active safety, automated driving, and parking-plus AI cockpit capabilities-scaling from Level 2+ to the highest levels of autonomy for large-scale production.	NVIDIA	Pytorch, MXNet, TensorFlow, Chainer	TensorRT, ONNX	NVIDIA DRIVE Hyperion, NVIDIA DRIVE Orin, NVIDIA DRIVE Thor	Autonomous vehicle, incabin functions, Driver monitoring	Active safety, Highway driving, Urban Driving & Parking, Cockpit
3	More customers, across a diverse set of industries, choose AWS compared to any other cloud to build, train, and deploy their machine learning (ML) applications. AWS delivers the broadest choice of powerful compute, high speed networking, and scalable high performance storage options for any ML project or application.	AWS	TensorFlow, PyTorch, and Apache MXNet.	TensorRT, Amazon EC2 InF1 Inference on Cloud	Amazon EC2 with custom silicon such as AWS Graviton (CPU) and AWS Inferentia	Autonomous driving (AD) and Advanced Driving Assistance Systems (ADAS)	Amazon Athena, Amazon Elastic Search, DynamoDB, FSx for Lustre

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 12 , December - 2025

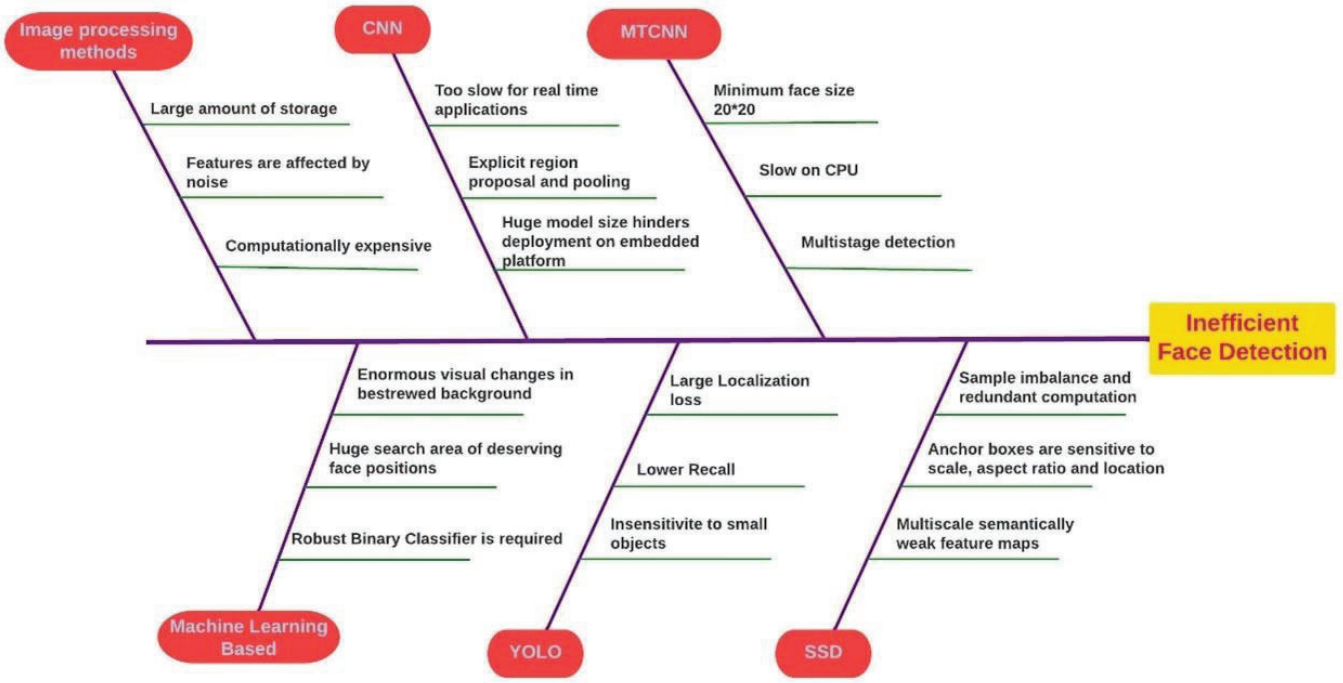

Fig. 11. Inefficient face detection affected by network architecture and performance metrics.

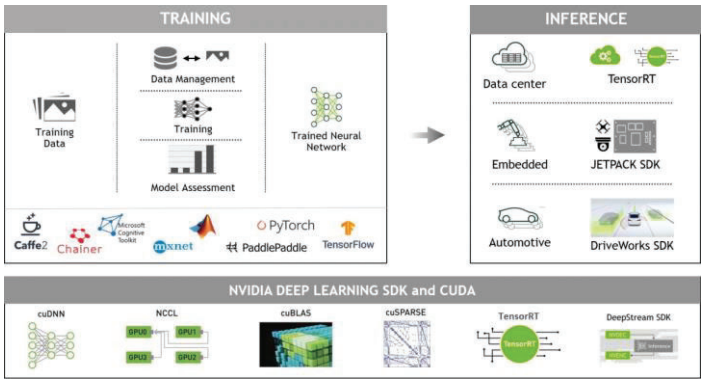

Fig. 12. NVIDIA GPU accelerated Deep Learning frameworks.

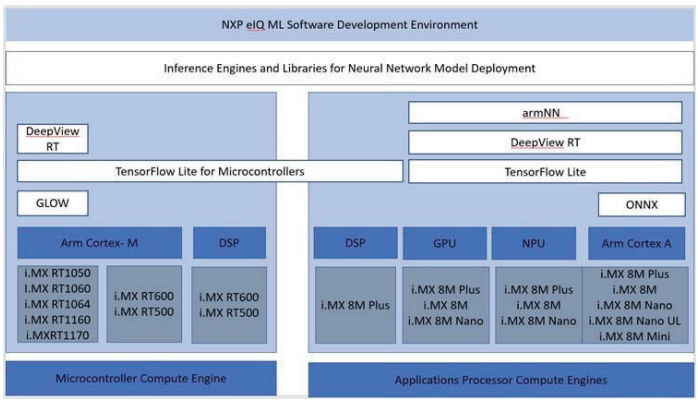

Fig. 13. NXP Software Development for Machine Learning

E. Interpretable Deep Learning models

Deep learning has set a performance benchmark on challenging face detection datasets, there many question that are unanswered. What is the underlying mechanism for the model? What is the relation between number of parameters and accuracy? How to attain maximum accuracy with minimal change in hyperparameters? What is effect of number of epochs and loss function in attaining a global minimum? We can monitor some of these features during training by using TensorFlow logs, yet a concise understanding about the interpretation of different components will yield a better face detection model.

VI. Conclusion

We have surveyed different modules of end-to-end real world deep learning application development impinged on diverse face detection architectures which have gained popularity on various benchmarking datasets. We have discussed the evolution stages for face detections methods and the critical improvements over predecessor ones. We have partitioned these models in various categories depending on architecture such as CNN, SSD, Yolo, and others, and have reported their major contributions for boosting the detection performance. An overview of practical software packages highlighting the package size, benefits and, drawbacks enabling the developer to curate appropriate algorithm for their application. We have summarized the need to utilize cloud instances and machine learning ADAS development tools due emergence of computer vision systems in driver monitoring systems from security aspect. Further, we have presented the challenges and oppor-

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 12 , December - 2025

tunities for face detection systems.

ACKNOWLEDGMENT

The authors express sincere gratitude to Center of Excel- lence in Signal and Image Processing, COEP Tech. Univer- sity, Pune, India and Fourfront Private Ltd, Pune, India for providing the facilities and funding for conducting the survey.

REFERENCES

[1] C. Machinery, "Computing machinery and intelligence-am turing," *Mind*, vol. 59, no. 236, p. 433, 1950.
[2] M. C. Burl, T. K. Leung, and P. Perona, "Face localization via shape statistics," in *Internatational Workshop on Automatic Face and Gesture Recognition*. University of Zurich, 1995, pp. 154–159.
[3] S.-H. Jeng, H. Y. M. Liao, C. C. Han, M. Y. Chern, and Y. T. Liu, "Facial feature detection using geometrical face model: An efficient approach," *Pattern recognition*, vol. 31, no. 3, pp. 273–282, 1998.
[4] S. Tsekeridou and I. Pitas, "Facial feature extraction in frontal views using biometric analogies," in *9th European Signal Processing Conference (EUSIPCO 1998)*. IEEE, 1998, pp. 1–4.
[5] K. C. Yow and R. Cipolla, "Feature-based human face detection," *Image and vision computing*, vol. 15, no. 9, pp. 713–735, 1997.
[6] C. Garcia and G. Tziritas, "Face detection using quantized skin color regions merging and wavelet packet analysis," *IEEE Transactions on multimedia*, vol. 1, no. 3, pp. 264–277, 1999.
[7] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2. Ieee, 1999, pp. 1150–1157.
[8] H. A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 20, no. 1, pp. 23–38, 1998.
[9] C. Garcia and M. Delakis, "Convolutional face finder: A neural architecture for fast and robust face detection," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 26, no. 11, pp. 1408–1423, 2004.
[10] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, vol. 1. Ieee, 2001, pp. I–I.
[11] Y.-Q. Wang, "An analysis of the viola-jones face detection algorithm," *Image Processing On Line*, vol. 4, pp. 128–148, 2014.
[12] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," *Lecture notes in computer science*, vol. 3951, pp. 404–417, 2006.
[13] K. Vikram and S. Padmavathi, "Facial parts detection using viola jones algorithm," in *2017 4th international conference on advanced computing and communication systems (ICACCS)*. IEEE, 2017, pp. 1–4.
[14] L. Cuimei, Q. Zhiliang, J. Nan, and W. Jianhua, "Human face detection algorithm via haar cascade classifier combined with three additional classifiers," in *2017 13th IEEE International Conference on Electronic Measurement & Instruments (ICEMI)*. IEEE, 2017, pp. 483–487.
[15] J. Huang, Y. Shang, and H. Chen, "Improved viola-jones face detection algorithm based on hololens," *EURASIP Journal on Image and Video Processing*, vol. 2019, no. 1, pp. 1–11, 2019.
[16] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1. Ieee, 2005, pp. 886–893.
[17] N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," in *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part II 9*. Springer, 2006, pp. 428–441.
[18] Q. Zhu, M.-C. Yeh, K.-T. Cheng, and S. Avidan, "Fast human detection using a cascade of histograms of oriented gradients," in *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, vol. 2. IEEE, 2006, pp. 1491–1498.
[19] D. E. King, "Max-margin object detection," *arXiv preprint arXiv:1502.00046*, 2015.
[20] X. Wang, T. X. Han, and S. Yan, "An hog-lbp human detector with partial occlusion handling," in *2009 IEEE 12th international conference on computer vision*. IEEE, 2009, pp. 32–39.
[21] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision– ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer, 2016, pp. 21–37.
[22] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, and S. Z. Li, "S3fd: Single shot scale-invariant face detector," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 192–201.
[23] J. Zhang, X. Wu, S. C. Hoi, and J. Zhu, "Feature agglomeration networks for single stage face detection," *Neurocomputing*, vol. 380, pp. 180–189, 2020.
[24] M. Najibi, P. Samangouei, R. Chellappa, and L. S. Davis, "Ssh: Single stage headless face detector," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4875–4884.
[25] J. Li, Y. Wang, C. Wang, Y. Tai, J. Qian, J. Yang, C. Wang, J. Li, and F. Huang, "Dsfd: dual shot face detector," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5060–5069.
[26] R. J. Wang, X. Li, and C. X. Ling, "Pelee: A real-time object detection system on mobile devices," *Advances in neural information processing systems*, vol. 31, 2018.
[27] X. Tang, D. K. Du, Z. He, and J. Liu, "Pyramidbox: A context-assisted single shot face detector," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 797–813.
[28] S. Zhang, L. Wen, H. Shi, Z. Lei, S. Lyu, and S. Z. Li, "Single-shot scale-aware network for real-time face detection," *International Journal of Computer Vision*, vol. 127, pp. 537–559, 2019.
[29] V. Bazarevsky, Y. Kartynnik, A. Vakunov, K. Raveendran, and M. Grundmann, "Blazeface: Sub-millisecond neural face detection on mobile gpus," *arXiv preprint arXiv:1907.05047*, 2019.
[30] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one- stage object detection," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9627–9636.
[31] Y. He, D. Xu, L. Wu, M. Jian, S. Xiang, and C. Pan, "Lffd: A light and fast face detector for edge devices," *arXiv preprint arXiv:1904.10633*, 2019.
[32] Z. Li, X. Tang, J. Han, J. Liu, and R. He, "Pyramidbox++: High performance detector for finding tiny face," *arXiv preprint arXiv:1904.00386*, 2019.
[33] S. Liu, D. Huang *et al.*, "Receptive field block net for accurate and fast object detection," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 385–400.
[34] L. Shi, X. Xu, and I. A. Kakadiaris, "Sanet: Smoothed attention network for single stage face detector," in *2019 International Conference on Biometrics (ICB)*. IEEE, 2019, pp. 1–7.
[35] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.
[36] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua, "A convolutional neural network cascade for face detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 5325– 5334.
[37] H. Jiang and E. Learned-Miller, "Face detection with the faster r-cnn," in *2017 12th IEEE international conference on automatic face & gesture recognition (FG 2017)*. IEEE, 2017, pp. 650–657.
[38] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.
[39] S. Yang, P. Luo, C.-C. Loy, and X. Tang, "From facial parts responses to face detection: A deep learning approach," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3676–3684.
[40] S. Yang, P. Luo, C. C. Loy, and X. Tang, "Faceness-net: Face detection through deep facial part responses," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 8, pp. 1845–1859, 2017.
[41] C. Zhu, Y. Zheng, K. Luu, and M. Savvides, "Cms-rcnn: contextual multi-scale region-based cnn for unconstrained face detection," *Deep learning for biometrics*, pp. 57–79, 2017.
[42] R. Ranjan, V. M. Patel, and R. Chellappa, "Hyperface: A deep multi- task learning framework for face detection, landmark localization, pose estimation, and gender recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 1, pp. 121–135, 2017.
[43] D. Chen, G. Hua, F. Wen, and J. Sun, "Supervised transformer network for efficient face detection," in *Computer Vision–ECCV 2016: 14th*

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 12 , December - 2025

European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14. Springer, 2016, pp. 122–138.

[44] Y. Wang, X. Ji, Z. Zhou, H. Wang, and Z. Li, "Detecting faces using region-based fully convolutional networks," *arXiv preprint arXiv:1709.05256*, 2017.

[45] K. Zhang, Z. Zhang, H. Wang, Z. Li, Y. Qiao, and W. Liu, "Detecting faces using inside cascaded contextual cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 3171–3179.

[46] S. Yang, Y. Xiong, C. C. Loy, and X. Tang, "Face detection through scale-friendly deep convolutional networks," *arXiv preprint arXiv:1706.02863*, 2017.

[47] C. Zhang, X. Xu, and D. Tu, "Face detection using improved faster rcnn," *arXiv preprint arXiv:1802.02142*, 2018.

[48] H. Wang, Z. Li, X. Ji, and Y. Wang, "Face r-cnn," *arXiv preprint arXiv:1706.01061*, 2017.

[49] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, and S. Z. Li, "Faceboxes: A cpu real-time face detector with high accuracy," in *2017 IEEE International Joint Conference on Biometrics (IJCB)*. IEEE, 2017, pp. 1–9.

[50] W. Wu, Y. Yin, X. Wang, and D. Xu, "Face detection with different scales based on faster r-cnn," *IEEE transactions on cybernetics*, vol. 49, no. 11, pp. 4017–4028, 2018.

[51] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A review of yolo algorithm developments," *Procedia Computer Science*, vol. 199, pp. 1066–1073, 2022.

[52] D. Garg, P. Goel, S. Pandya, A. Ganatra, and K. Kotecha, "A deep learning approach for face detection using yolo," in *2018 IEEE Punecon*. IEEE, 2018, pp. 1–4.

[53] W. Yang and Z. Jiachun, "Real-time face detection based on yolo," in *2018 1st IEEE international conference on knowledge innovation and invention (ICKII)*. IEEE, 2018, pp. 221–224.

[54] D. Qi, W. Tan, Q. Yao, and J. Liu, "Yolo5face: Why reinventing a face detector," in *European Conference on Computer Vision*. Springer, 2022, pp. 228–244.

[55] W. Chen, H. Huang, S. Peng, C. Zhou, and C. Zhang, "Yolo-face: a real-time face detector," *The Visual Computer*, vol. 37, pp. 805–813, 2021.

[56] D. T. Nguyen, T. N. Nguyen, H. Kim, and H.-J. Lee, "A high- throughput and power-efficient fpga implementation of yolo cnn for object detection," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 8, pp. 1861–1873, 2019.

[57] S. Zhang, J. Cao, Q. Zhang, Q. Zhang, Y. Zhang, and Y. Wang, "An fpga-based reconfigurable cnn accelerator for yolo," in *2020 IEEE 3rd International Conference on Electronics Technology (ICET)*. IEEE, 2020, pp. 74–78.

[58] R. Huang, J. Pedoeem, and C. Chen, "Yolo-lite: a real-time object detection algorithm optimized for non-gpu computers," in *2018 IEEE international conference on big data (big data)*. IEEE, 2018, pp. 2503–2510.

[59] Y. Zhu, H. Cai, S. Zhang, C. Wang, and Y. Xiong, "Tinaface: Strong but simple baseline for face detection," *arXiv preprint arXiv:2011.13183*, 2020.

[60] J. Deng, J. Guo, E. Ververas, I. Kotsia, and S. Zafeiriou, "Retinaface: Single-shot multi-level face localisation in the wild," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 5203–5212.

[61] K. Lin, H. Zhao, J. Lv, C. Li, X. Liu, R. Chen, and R. Zhao, "Face detection and segmentation based on improved mask r-cnn," *Discrete dynamics in nature and society*, vol. 2020, pp. 1–11, 2020.

[62] S. Zhang, C. Chi, Z. Lei, and S. Z. Li, "Refineface: Refinement neural network for high performance face detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 11, pp. 4008– 4020, 2020.

[63] K. Liu, H. Tang, S. He, Q. Yu, Y. Xiong, and N. Wang, "Performance validation of yolo variants for object detection," in *Proceedings of the 2021 International Conference on bioinformatics and intelligent computing*, 2021, pp. 239–243.

[64] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271.

[65] P. Adarsh, P. Rathi, and M. Kumar, "Yolo v3-tiny: Object detection and recognition using one stage improved model," in *2020 6th interna- tional conference on advanced computing and communication systems (ICACCS)*. IEEE, 2020, pp. 687–694.

[66] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Op- timal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[67] C. Gupta, N. S. Gill, P. Gulia, and J. M. Chatterjee, "A novel finetuned yolov6 transfer learning model for real-time object detection," *Journal of Real-Time Image Processing*, vol. 20, no. 3, p. 42, 2023.

[68] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 7464–7475.

[69] M. Khan, S. Chakraborty, R. Astya, and S. Khepra, "Face detection and recognition using opencv," in *2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*. IEEE, 2019, pp. 116–119.

[70] C. Zhang and Z. Zhang, "Improving multiview face detection with multi- task deep convolutional neural networks," in *IEEE Winter Conference on Applications of Computer Vision*. IEEE, 2014, pp. 1036–1041.

[71] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE signal processing letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[72] J. Xiang and G. Zhu, "Joint face detection and facial expression recog- nition with mtcnn," in *2017 4th international conference on information science and control engineering (ICISCE)*. IEEE, 2017, pp. 424–427.

[73] P. Hu and D. Ramanan, "Finding tiny faces," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 951–959.

[74] C. Zhu, R. Tao, K. Luu, and M. Savvides, "Seeing small faces from robust anchor's perspective," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5127–5136.

[75] V. Albiero, X. Chen, X. Yin, G. Pang, and T. Hassner, "img2pose: Face alignment and detection via 6dof, face pose estimation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 7617–7627.

[76] T. Baltrus˘aitis, P. Robinson, and L.-P. Morency, "Openface: an open source facial behavior analysis toolkit," in *2016 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 2016, pp. 1–10.

[77] G. Ghiasi and C. C. Fowlkes, "Occlusion coherence: Detecting and localizing occluded faces," *arXiv preprint arXiv:1506.08347*, 2015.

[78] X. Shi, S. Shan, M. Kan, S. Wu, and X. Chen, "Real-time rotation- invariant face detection with progressive calibration networks," in *Pro- ceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2295–2303.

[79] J. Chen and X. Ran, "Deep learning with edge computing: A review," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1655–1674, 2019.

[80] H. Hua, Y. Li, T. Wang, N. Dong, W. Li, and J. Cao, "Edge comput- ing with artificial intelligence: A machine learning perspective," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–35, 2023.

[81] G. Lacey, G. W. Taylor, and S. Areibi, "Deep learning on fpgas: Past, present, and future," *arXiv preprint arXiv:1602.04283*, 2016.

Snehal Patil is a Research Scholar at COEP Technological University, Pune, India. She has completed her MTech from College of Engineering, Pune. Her field of interests include machine learning and application deployment over cloud. She has a teaching experience of 9 years. She has successfully completed Funded Research Project in field of esmbedded systems.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 12 , December - 2025

Prashant Bartakke is currently work- ing as Dean at School of Electrical and Communication Engineering at COEP Technological University, Pune, India. He completed Ph.D. in E & TC Engg., Uni- versity of Pune in December 2011, Topic of Research: "Stochastic Modeling of Textures for Analysis and Synthesis". His areas of interest are Embedded System Design, RTOS, Ad- vanced Embedded Architecture, Image Processing and Pattern Recognition. He has handled diverse funded projects such as Machine Vision System for Material Surface Inspection in Industrial Applications, To deploy detection and classification of the soldering defects in PCB manufacturing on Intel atom platform using images acquired by stereo microscope, and Real time Vehicle Tracking Information System : Supported by position logger using Raspberry Pi.

Mukul Sutaone is currently working as Director at IIIT, Allahabad, India. He has an experience of 25+ years. He pursued Ph. D. (Electronics and Telecomm.)- University of Pune and College of Engineering, Pune in 2005. He is recipient of prestigious Prof. SVC Aiya Award from Institution of Electronics and Telecommunication (IETE) for Best Teacher in Telecomm. Education, in December 2010. Areas of Interests in Teaching and Research: Multimedia Signal Processing, Communication Network technologies, VLSI Design for Signal Processing Applications. He successfully completed three funded research projects in the areas of Fiber Optic Communication, Iris Image Analysis for clinical diagnosis and PCB design and fabrication, worth INR 4.7 Millions. He is Principal Investigator and Conceiver of the Multidisciplinary "Center of Excellence in Signal and Image Processing" at COEP, a Project worth INR 50 lakhs.

Rajesh Chavan has completed MBA and has 35 years of Industry experience with Transasia Bio-medical Ltd and Honeywell Pvt Ltd. His expertise include Electronics Product Design, New Product Introduction and Design for Six-sigma. He has versatile experience in domains like Vibration Monitoring, Industrial Automation, Office Automation , Bio-Chemistry Analysers, optical reflection densitometry.