IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Face Detection In A Video Using Big Data Tools And Computer Vision Techniques

[1]RamaDevi Gunnam, [2]Suneetha Neerumalla, [3]Venkata Lakshmi Gayathri Pabbisetty, [4]Koteswara Rao Nadikoti, [5]Kahar Moula Shaik

[1]Assistant Professor, [2]Student, [3]Student, [4]Student, [5]Student

[1]Computer Science and Engineering,

[1]Vasireddy Venkatadri Institute of Technology, Guntur, India

Abstract: Face detection, a process of identifying faces within video content, has traditionally faced limitations in precision. Leveraging big data techniques can enhance speed and efficiency. Employing Apache Spark, capable of running among several nodes simultaneously, facilitates the conversion of individual video frames into an analyzable format. Utilizing the YOLO Face Detection model, rooted in deep learning, enables the detection of faces within each frame. Deep Face will then be utilized to pinpoint the target face and extract all instances across multiple frames for monitoring purposes.

***Index Terms* - Apache Spark, Big Data, Computer Vision, Deep Face, Face Detection, YOLOv8.**

I. Introduction

The application of face detection in video, leveraging big data tools and computer vision techniques, is a fascinating intersection of intensive handling of data and advanced image analysis. Face detection, a crucial task in computer vision, involves the identification and localization of human faces in still or moving images or video frames. This process plays a vital role in various applications, such as security monitoring, emotion analysis, and facial recognition.

To achieve face detection in videos, algorithms analyses the visual elements of an image or video. Given the considerable volume of data involved in video streams, using big data solutions becomes imperative. Apache Spark, a notable example, proves invaluable due to is scalability, fault tolerance, and distributed processing capabilities.

Furthermore, the efficiency of face detection algorithms is critical in real-world scenarios. They must handle dynamic changes in lighting conditions, varying facial expressions, and diverse backgrounds. Robust algorithms contribute to the accuracy and reliability of applications relying on face detection.

OpenCV, an abbreviation for the Computer Vision OpenSource Library, stands as an open-source software library for computer vision and machine learning tasks. It was initially developed by Intel in 1999 and has since become one of the most widely used libraries for real-time computer vision applications. Here's a breakdown of its key features and capabilities. OpenCV provides a vast array of tools and algorithms for image processing tasks such as filtering, edge detection, morphological operations, and color space manipulation. As face detection applications continue to expand, the integration of real-time processing becomes crucial. Real-time face detection is essential in scenarios such as surveillance, where quick and accurate identification is paramount. This necessitates not only efficient algorithms but also the utilization of parallel processing and optimized hardware for rapid decision-making. Moreover, privacy concerns have become increasingly relevant in the deployment of face detection systems. Striking a balance between the benefits of facial analysis and respecting individual privacy is a critical consideration. Ethical use and responsible handling of facial data are integral aspects that require attention as these technologies advance.

II. LITERATURE REVIEW

[1] Face detection in real-world scenarios poses challenges due to variations in expressions, lighting, and occlusions. Deep learning techniques have shown remarkable success in addressing these challenges by leveraging complex strategies. The proposed deep cascaded multi-task framework exploits correlations between tasks, enhancing performance in face detection and landmark localization. Further improvements and applications of the proposed method are envisaged, leveraging the insights gained from experimentation and analysis. [2] This paper provides an extensive overview of face detection methods, categorizing them into feature-based and image-based approaches, with a focus on the burgeoning role of neural networks (NNs). It highlights the historical evolution of face detection algorithms and their recent advancements propelled by computational resources. The comparative evaluation among different algorithms elucidates their strengths and limitations, shedding light on their applicability in diverse scenarios. Notably, NN models emerge as high-performing algorithms in face detection, leveraging hardware developments to tackle challenges like occlusion and complex backgrounds. The survey extends beyond face detection, exploring the extensive utilization of these algorithms in fault diagnosis, EEG data analysis, and various pattern recognition tasks. The research underscores the imperative of advancing face detection technology to mitigate false positives and unlock its full potential in critical domains.

[3] The study presents a novel approach to physical security using facial recognition technology, integrating Haar Cascade for face detection and LBPH for face recognition. The proposed security system prototype utilizes a microcomputer, Raspberry Pi 4, as the central processing unit to analyze camera feed and control access to the secured area. Overall, the proposed security system offers a promising solution for industries seeking enhanced access control and surveillance capabilities, leveraging facial recognition technology for improved security protocols and incident response mechanisms. [4] The proposed correlation-based approach utilizes response maps from CNN models to improve face detection in video sequences, focusing on aligning face features more accurately in the embedding space. Leveraging automatic neuron selection and response map generation techniques, the framework aims to identify optimal face locations for more consistent and interpretable results. Overall, the proposed correlation-based approach offers a novel and effective solution for enhancing face recognition performance in video sequences, with potential applications in various domains requiring robust and accurate facial recognition capabilities.

[5] Face recognition technology, with its capability to capture and match facial features, has witnessed significant advancements, driven by the robustness of deep learning models and its widespread application across various sectors. The proposed AB-FR model integrates an attention based convolutional neural network (CNN) to enhance feature extraction from facial images, thereby improving recognition performance and robustness. By incorporating bidirectional long short-term memory (BiLSTM), the model captures temporal and geometric features across different images of the same individual, addressing the limitations of traditional deep learning-based methods in considering timing characteristics. The utilization of cross-entropy loss function facilitates efficient model optimization and classification, resulting in improved accuracy and training effectiveness. [6] The integration of face mesh technology with deep neural networks represents a promising approach for robust face detection and recognition, offering versatility across varying conditions such as illumination changes and non-frontal poses. Leveraging the Labeled Wild Face (LWF) dataset and real-time image captures, the proposed model demonstrates significant advancements in accurately identifying individuals, with a commendable accuracy rate of 94.23% achieved for face recognition tasks. This research contributes to the evolving landscape of computer vision and deep learning, providing practical solutions for applications ranging from access control in secure areas to database-driven identification systems, thereby addressing key challenges in face detection and recognition.

[7] This paper offers a comprehensive overview of existing face recognition methods and the factors influencing facial recognition accuracy, addressing the heightened demand for biometric security systems in safeguarding against fraud and theft. It delves into the impact of factors such as aging, pose variation, partial occlusion, illumination, and facial expressions on recognition accuracy, while also exploring techniques aimed at mitigating these effects. [8] The F-DR Net represents a pioneering approach in the realm of face multi-task analysis, offering a unified solution for both face detection and recognition within a single network architecture, addressing the challenges posed by these tasks. Extensively evaluated on benchmark datasets such as Labeled Faces in the Wild (LFW) and Face Detection Datasets and Benchmark (FDDB), the F-DR Net demonstrates superior performance in terms of detection and recognition accuracy while maintaining

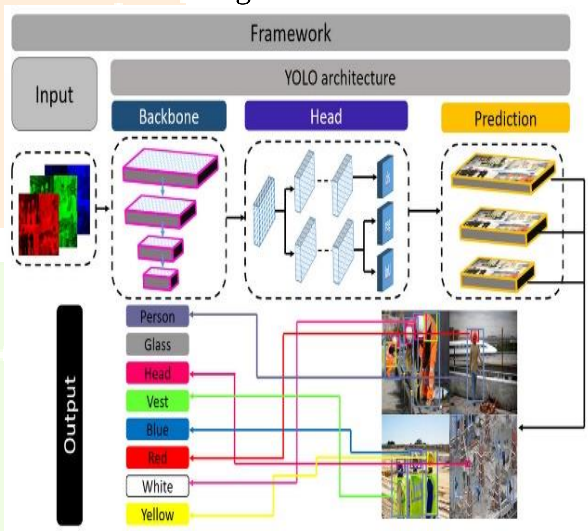
faster processing speeds compared to existing methods, underscoring its efficacy and practical utility in real-world scenarios. [9] This study presents a new face recognition attendance system leveraging real-time video processing, showcasing its efficacy in enhancing attendance monitoring in educational settings and potentially boosting student engagement and institutional efficiency.

[10] This paper presents a thorough review of current research trends in video-based face recognition and retrieval, highlighting the shift from traditional still-image approaches to leveraging the rich and redundant information provided by video data for more accurate and robust recognition in unconstrained environments. [11] This paper presents an experimental surveillance system that addresses the challenges of robust face recognition in real-time applications, emphasizing the utilization of Viola-Jones face detection coupled with shape-based feature extraction to ensure resilience to variations in illumination, scale, pose, and expression, thus enhancing both detection and recognition accuracy. [12] This paper presents a robust approach for face detection in multiple views and landmark localization using Multi-task Cascaded Convolutional Networks (MTCNN), demonstrating high accuracy in both frontal and non-frontal face detection scenarios. The proposed method enhances detection accuracy, particularly for large-angle faces, offering a comprehensive solution for complex environments.

III. PROPOSED METHODOLOGY

3.1 Understanding Yolov8 Architecture

Fig 1:Yolov8 Architecture



YOLOv8, or You Only Look Once version 8, is an object detection algorithm that belongs to the YOLO (You Only Look Once) family of models. YOLO models are known for their real-time object detection capabilities, meaning they can quickly and accurately identify objects within an image or a video frame.

Backbone Network: YOLOv8 uses a backbone network, often based on CSPDarknet53, which is an extended version of Darknet, a deep neural network designed for object detection. The backbone extracts features from the input image.

Feature Pyramid Network (FPN): YOLOv8 employs a Feature Pyramid Network to capture features at different scales. This helps in detecting objects of varying sizes within an image.

Detection Head: The detection head in YOLOv8 is tasked with predicting both bounding boxes and the probabilities of various classes. It uses a set of convolutional layers to process the feature maps from the backbone and FPN, producing predictions for each grid cell.

Anchor Boxes: YOLOv8 utilizes anchor boxes to improve bounding box prediction accuracy. These anchor boxes are predefined sizes that help the model better adapt to the scale of different objects in the image.

Output Format: The final output of YOLOv8 is a set of bounding boxes, each associated with a specific class and its confidence score. The bounding boxes are predicted for different grid cells and anchor boxes at multiple scales.

Training: YOLOv8 is trained using a combination of labeled images and their corresponding bounding box annotations. The model is optimized using techniques like stochastic gradient descent with backpropagation.

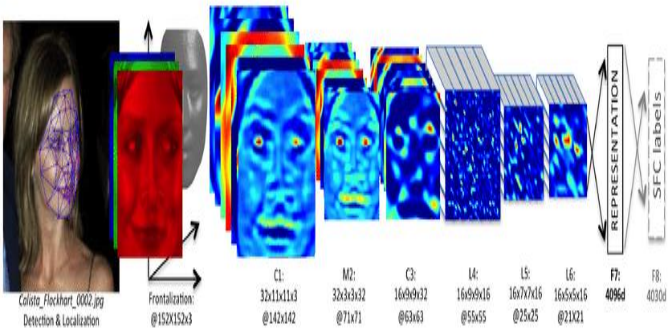
Variants: YOLOv8 has different variants, including YOLOv8-S, YOLOv8-M, YOLOv8-L, and YOLOv8-X, with varying model sizes and computational demands. Users can choose a variant based on their specific requirements regarding speed and accuracy.

Efficiency and Real-time Processing: YOLOv8 is known for its efficiency, making it suitable for real-time object detection applications. It processes the entire image in one forward pass, eliminating the need for complex post-processing steps.

It's important to note that YOLOv8 builds upon the success of its predecessors and incorporates improvements to enhance detection accuracy and speed. The architecture's flexibility and balance between accuracy and speed make it a popular choice for various computer vision tasks.

3.2 Understanding Deep Face

Fig 2: Deep Face Architecture



"Deep Face" refers to a deep learning-based face recognition system developed by Facebook's AI Research (FAIR) lab. It gained significant attention when it was introduced in 2014 due to its high accuracy in identifying faces across large datasets. Here's an overview:

Architecture: Deep Face utilizes a deep convolutional neural network (CNN) architecture, which is trained on a massive dataset of labeled face images. The network consists of multiple layers that learn hierarchical representations of facial features, enabling it to recognize faces with high accuracy.

Face Verification: Deep Face is primarily designed for face verification tasks, which involve determining whether two face images belong to the same person or not. It accomplishes this by encoding facial images into a compact representation (embedding) in a high-dimensional space, where the similarity between two embeddings can be measured using techniques like cosine similarity or Euclidean distance.

Training Data: Deep Face was trained on a large-scale dataset called the "Social Face Classification" dataset, which contains millions of labeled face images belonging to thousands of individuals. This extensive training dataset helped Deep Face learn robust representations of facial features, enabling it to generalize well to unseen faces.

Performance: Deep Face achieved state-of-the-art performance on various face recognition benchmarks, surpassing previous methods in terms of accuracy and scalability. In particular, it demonstrated high accuracy even in challenging conditions such as variations in pose, lighting, and facial expression.

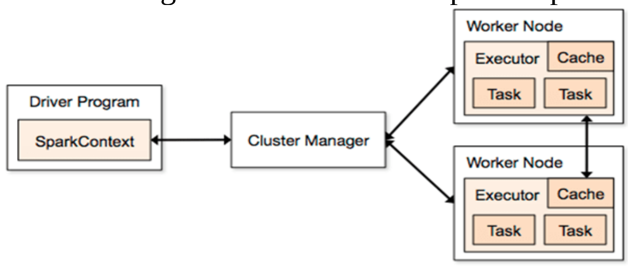
Applications: Deep Face has various applications in areas such as biometric authentication, surveillance, social media, and photo tagging. It can be used to automatically tag friends in photos on social media platforms, verify users' identities for secure access control, and assist in law enforcement efforts for identifying individuals in surveillance footage.

Privacy and Ethical Considerations: The deployment of facial recognition systems like Deep Face raises concerns about privacy, surveillance, and potential misuse of personal data. Critics argue that such technologies can infringe on individuals' privacy rights and exacerbate surveillance issues if not regulated properly. Therefore, ethical considerations and regulatory frameworks are crucial in ensuring responsible creation and use of facial recognition technology.

Overall, Deep Face represents a significant advancement in the field of face recognition, demonstrating the power of deep learning techniques for accurately identifying and verifying individuals in images. However, it also underscores the importance of addressing privacy and ethical concerns associated with the widespread adoption of facial recognition technology.

3.3 Understanding Apache Spark

Fig 3: Architecture Of Apache Spark



Fast and versatile, Apache Spark is a cluster computing platform made. for distributed data processing. It offers a programming interface with implicit fault tolerance and data parallelism for entire clusters. Spark was created in Scala and is compatible with many programming languages, such as java, Python and R, so a broad spectrum of developers can use it.

There are lot of features in spark. When compared to other conventional data processing technologies, Spark is renowned for its extraordinary speed. By using in-memory processing, it lessens the requirement for a lot of disk input/output. Faster processing speeds are achieved by optimizing job execution through the usage of the Directed Acyclic Graph (DAG) execution.

A high-level API in Java, Scala, Python and R is provided by Spark, which streamlines the development process. Additionally, it offers interactive shell functionality so that code snippets can be examined and tested before being integrated into more complex workflows

There is high adaptability in Apache Spark. Resilient distributed datasets and lineage information are two ways that spark provides fault tolerance. Spark can recompute an RDD using lineage information in the event that a partition is lost, guaranteeing data integrity and dependability.

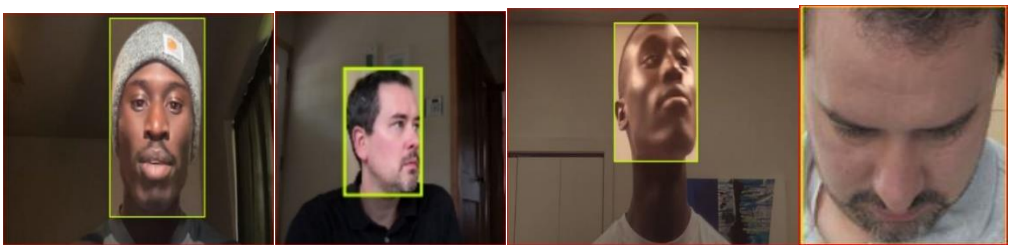
IV. IMPLEMENTATION

4.1 Dataset Description:

4.1.1 Dataset Preparation:

The dataset consists of images containing faces, with each face annotated with bounding boxes. This dataset name is FaceDetection18.

Fig 4: Dataset Images



Dataset split:

Train Set: 87% (2871 images)

Valid Set: 8% (267 images)

Test Set: 4% (145 images)

Each image is annotated with bounding boxes indicating the location of faces.

4.1.2 Data Preprocessing:

Dataset is preprocessed as

Auto-Orient: Applied

Resize: Stretch to 640*649

4.1.3 Data Augmentation:

Table 1

Parameters	Value
90deg Rotate	Clockwise, Counter-Clockwise,Upside Down
Flip	Horizontal, Vertical
Crop	0% Minimum Zoom, 25% Maximum Zoom
Gray Scale	Apply to 5% of images
Hue	Between -20° and +20°
Saturation	Between -10% and +10%
Brightness	Between -10% and +10%
Exposure	Between -10% and +10%
Blur	Up to 1px
Cutout	5boxes with 2% size each

4.2 Model Training in Face Detection:

The YOLOv8 model architecture is used for face detection tasks.

4.2.1 The training process involves:

Initializing the YOLOv8 model with pre-trained weights (yolov8s.pt). Setting up data loaders to load the training dataset. Training the model on the training dataset for multiple epochs.

4.2.2 Model Validation:

After training, the trained model is validated on a separate validation set to evaluate its performance. The validation process includes: Initializing the validation module with the trained model and validation dataset. Computing various metrics using confusion matrix for object detection tasks. Evaluating the model's performance on the validation dataset and recording the evaluation results.

4.2.3 Prediction on Unseen Data:

Once the model is trained and validated, it can be used to make predictions on unseen data, such as images or videos containing faces.

4.3 Face Verification using Deep Faces

Load the pretrained Deep Face model and Here input is given as detected faces file and target image to get instances of input image in video. Here the algorithm extract the features from detected faces and match features against known identities.

Fig 5: Deep Face Example

Verify function under the deep face interface is used for face recognition. Modern face recognition pipelines consist of 4 stages: detect, align, represent and verify. Deep Face handles all these common stages in the background.

Each call of verification builds a face recognition model from scratch and this is a costly operation. If you are going to verify multiple faces sequentially, then you should pass an array of faces to verification function to speed the operation up. In this way, complex face recognition models will be built once.
Sample code:

```
from deepface import DeepFace
result=DeepFace.verify("img1.jpg","img2.jpg")
print("Is verified: ", result["verified"])
```

Combining the processed frames into Video:

Here, in these combining frames we take the matched faces of frames from deep face module and combine them using Moviepy module.

Applying Big data:

Applied big data for parallel processing of frames using Apache spark. Input image and video is processed and the frames of the video will be passed to Apache spark and it will be distributing the frames to processors and will be processing parallelly and finally returning the frames which match with the image.

V. RESULTS AND DISCUSSIONS

In this approach face detection using yolov8 is used and input is given as video along with target image and model is trained by using face dataset this gives a value of 90.1 mAp (**Mean Average precision**) value and it is used as Train Set: 87% (2871 images), Valid Set: 8% (267 images), Test Set: 4% (145 images) and it is also preprocessed with auto-orient is applied along with resized to make stretch to 640x640. For face verification we used the deep face to match faces from face detection output with target input image and output frames are collected and merged into video using Moviepy module and frames processing is done using Apache spark. Output will be video consisting of target image. Incorporating Big data technology like Apache Spark for parallel processing of frames. Input image and video is processed and the frames of the video will be passed to Apache spark and it will be distributing the frames to processors and will be processing parallelly and finally returning the frames which match with the image.

Fig 6: Output Image1

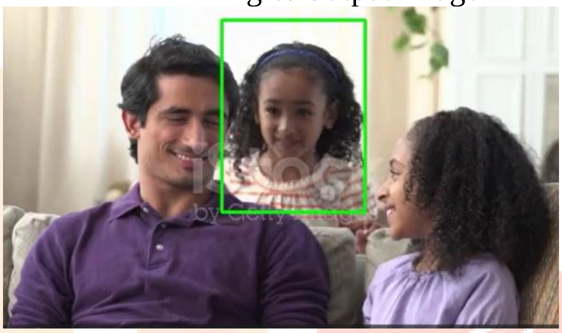


Fig 7: Output Image2

VI. CONCLUSIONS

Computer Vision is the subfield of Artificial Intelligence, where computers are trained to process the image and extract the important features from the images or videos. Applications of computer vision are object detection, face recognition, medical diagnosis, etc. In this paper we emphasize the important role of yolov8 for face detection and deep face for face verification. Finally, we assessment and compared recent literature reviews that use Opencv to detect the human face.

VII. LIMITATIONS AND FUTURE SCOPE

In this approach we used to detect faces from video and to recognize the target image with detected faces in these there may be chance that it may not recognize images with some occlusions.Future work includes the matching the image with masks and also as include the working with more processors for faster execution.

VIII. ACKNOWLEDGMENTS

The authors would like to thank the CSE Department and extend our sincere appreciation for the assistance and guidance provided throughout the completion of this paper. Our gratitude goes to our project guide and project coordinator for their invaluable support and advice throughout the project.

REFERENCES

[1] Yuxuan Liao, Zhenyu Tang, Jiehong Lei, Jiajia Chen, &Zhong Tang. (2022). Video Face Detection Technology and Its Application in Health Information Management System

[2] Supriya Priya R Asaithambi, Sitalakshmi Venkataraman, & Ramanathan Venkataraman. (2021). Proposed Big data architecture for facial recognition using machine learning

[3] Peng, B., & Gopalakrishnan, A. K. (2019). A Face Detection Framework Based on Deep Cascaded Full Convolutional Neural Networks. 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS).

[4] Hao, Z., Feng, Q., & Kaidong, L. (2018). An Optimized Face Detection Based on Adaboost Algorithm. 2018 International Conference on Information Systems and Computer Aided Education (ICISCAE).

[5] Wati, D. A. R., & Abadianto, D. (2017). Design of face detection and recognition system for smart home security application. 2017 2nd International Conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE).

[6] Hsu, H.-W., Wu, T.-Y., Wong, W. H., & Lee, C.-Y. (2018). Correlation-Based Face Detection for Recognizing Faces in Videos. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).

[7] Rong Xie ,Qingyu Zhang, Enyuan Yang, Qiang Zhu(2019).A method of small Face Detection Based on CNN.2019 4th international Conference on Computational Intelligence and Applications(ICCIA)

[8] Shivalila Hangaragi, Tripty Singh, NeelimaN(2023), Face Detection and Recognition Using Face Mesh and Deep Neural Network. International Conference on Machine Learning and Deep Neural Network.

[9] Shahina Anwarul,Susheela Dahiya(2020). A Comprehensive Review on Face Recognition Methods and Factors Affecting Facial Recognition Accuracy.

[10] Pang, L., Ming, Y., & Chao, L. (2018). F-DR Net:Face detection and recognition in One Net. 2018 14th IEEE International Conference on Signal Processing (ICSP).

[11] Hao Yang and Xiaofeng Han .Face Recognization Attendence System Based on Real time Video Processing (2020).

[12] HtetAung ,Alexander V.Bobkov,Nyan LinTun. Face detection in real time video using Yolo algorithm based on Vgg16 ConvolutionalNeural Network. 2021 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM).

[13] Caifeng Shan(2010) Face Recognition and Retrieval in Video.

[14] Davis, M., Popa, S., & Surlea, C. (2010). Real-Time Face Recognition from Surveillance Video. In Intelligent Video Event Analysis and Understanding (1st ed., pp. 155-194). (Studies in Computational Intelligence; Vol. 332). Springer

[15] Ma, M., & Wang, J. (2018). Multi-View Face Detection and Landmark Localization Based on MTCNN. 2018 Chinese Automation Congress (CAC).

[16] Cuimei, L., Zhiliang, Q., Nan, J., & Jianhua, W. (2017). Human face detection algorithm via Haar cascade classifier combined with three additional classifiers. 2017 13th IEEE International Conference on Electronic Measurement & Instruments (ICEMI).

[17] Payal Bose1, Samir K Bandhyopadhyay1 and Vishal Goyal(2020) Human Different Head Poses and Facial Expression Analysis using Principal Component Analysis. 2020 Xeno Journal of Biomedical Sciences.