Face Detection with Landmark using YOLO(v8)

Nehith Sai Vemulapalli
Amrita Vishwa Vidyapeetham
Kerala, India
AM.EN.U4AIE20053

Paladugula Pruthvi
Amrita Vishwa Vidyapeetham
Kerala, India
AM.EN.U4AIE20054

Guggilam Sai Prabhat
Amrita Vishwa Vidyapeetham
Kerala, India
AM.EN.U4AIE20032

Scrum Master Name & Signature **Product Owner Name & Signature**

Software Framework Activities

I. CHAPTER : COMMUNICATION

Team Communication:

In our project, effective team communication was the cornerstone of our success. As a tightly-knit team of three, we harnessed a variety of communication tools such as WhatsApp, Microsoft Teams, and Discord to maintain seamless connectivity throughout the project's duration. These platforms enabled us to exchange ideas, share updates, and stay in the loop with one another. Beyond the digital realm, we also made it a point to meet in person, fostering a collaborative environment where we could discuss various facets of our research. Right from the project's inception, we embarked on a journey of joint brainstorming sessions, idea sharing, and mutual decision-making to define our project's focal point. This proactive approach to communication facilitated the effective distribution of responsibilities.

Collaboration with Advisors and Mentors:

Our project mentor played a pivotal role in guiding us through the research process. Her guidance and support were invaluable, as she actively encouraged us to aim for the publication of our research findings in a conference paper. In our interactions with her, we had the opportunity to bounce around project ideas, share our insights, and tap into her wealth of knowledge. It was under her expert tutelage that we zeroed in on our project's subject matter. We were fortunate to engage in face-to-face discussions within the academic setting, a format that enhanced our understanding and provided a structured approach to guidance. Additionally, we maintained ongoing communication with her via platforms like WhatsApp, ensuring a continuous exchange of ideas and progress updates.

Literature Review:

A meticulous literature review process was instrumental in shaping our project. Each member of our team actively participated in this endeavor, dedicating substantial effort to the exhaustive review of relevant papers on IEEE Xplore. Our approach involved carefully assessing the strengths and weaknesses of the papers we selected, while also identifying potential research gaps. To streamline our collective efforts, we divided the literature review tasks amongst ourselves, ultimately converging on our findings through collaborative discussions. While we did refer to some existing articles, it became evident that the literature on our specific project was rather scarce. Thus, we relied significantly on our own research efforts, underscoring the importance of original contributions.

Model Training and Development:

For our project, we employed Yolo-v8 for Face Detection with Landmark identification. This advanced model streamlined precise face detection and landmark pinpointing on our dataset. However, extracting landmark coordinates was time-intensive, consuming nearly a day for comprehensive data processing. Given this duration, efficient communication became crucial within our team. As we navigated the complexities of Yolo-v8, the importance of teamwork and transparent communication was underscored, ensuring we tackled challenges and stayed in sync throughout the endeavor.

Report Writing and Publication:

Our journey through report writing and publication adhered closely to professional standards. To ensure our research paper met the required format, we drew upon IEEE templates available on the conference website, adopting LaTeX for document formatting. With painstaking attention to detail, we engaged in meticulous reviews of the paper, with a specific focus on proper referencing and accurate citation of images. Addressing concerns related to plagiarism, we embarked on multiple revisions, meticulously refining our content to align with the highest ethical publication standards. Furthermore, we resized images to comply with specified limits, a testament to our unwavering commitment to delivering a polished and compliant research paper.

II. Chapter : Planning

A. *Abstract :*

The advent of deep learning combined with computer vision has brought forth unparalleled advancements in facial detection and landmark identification. One pivotal player in this transformation has been the YOLO (You Only Look Once) series, setting new milestones in object detection methodologies. Our research is centered on harnessing the YOLOv8 model to optimize face detection processes. We incorporate the OpenCV library for image processing, enhancing detection fidelity through adjustable parameters like confidence and intersection over union (IoU) thresholds. A standout feature of our methodology is its innate ability to adjust to diverse image proportions. This is achieved by innovatively resizing and padding input images, which not only maintains consistency in detection but also augments accuracy. The proposed technique not only demarcates the face but also pinpoints facial landmarks, thus offering a comprehensive spatial map for each detected face. Preliminary results on benchmark datasets underscore the model's dual advantage of speed and precision. Our approach promises not only improved face detection but also paves the way for its amalgamation into expansive facial recognition and analytic platforms.

Task	Start Date	End Date
Literature Review	15-09-23	20-09-23
Data Collection And Preprocessing	20-09-23	21-09-23
Model Implementation	22-09-23	28-09-23
Experiments And Results	28-09-23	30-09-23
Report Writing	30-09-23	03-10-23
Review And Revisions	03-10-23	11-10-23
Conference Submission	18-10-23	

TABLE I: Project Timelines and Deadlines

B. *Budget :*

Our project's budget encompassed several critical aspects to ensure the successful execution of our research. These included:

*1) **Conference Fees :*** In our pursuit of sharing our research findings and contributing to the academic community, we aimed to present our work at a relevant conference. This might require us to pay registration fees for the conference. The conference fee would cost us around 9,000 rupees. This expense would be necessary to participate in the conference and submit our paper for review and presentation.

*2) **Miscellaneous Expenses :*** Our budget allocation also accounted for potential miscellaneous expenses that might arise during the course of the project. These miscellaneous expenses could include costs associated with acquiring specific software licenses, obtaining additional data for our experiments, or addressing unforeseen needs that might emerge during the project's lifecycle. The specific amount allocated for miscellaneous expenses was determined based on the project's evolving requirements and any unexpected contingencies.

The allocation of our project's budget was meticulously planned, taking into consideration quotations from relevant service providers and our project's specific needs. By budgeting for conference fees, and potential miscellaneous expenses, we ensured that we had the financial resources necessary to support our research and successfully achieve our project's objectives.

C. *Team Division :*

Our project team operated cohesively with a division of roles and responsibilities, ensuring the efficient execution of tasks and the attainment of our research goals. Despite our relatively small team, each member actively contributed to various facets of the project. Our team roles were as follows:

*1) **Project Lead :*** As the leader of our project, the Project Lead assumed the crucial role of overseeing the entire project's progression. Their responsibilities encompassed project management, ensuring that the project remained on schedule, coordinating the efforts of team members, and serving as a point of contact for communication within the team. The Project Lead played a pivotal role in maintaining the project's direction and alignment with its objectives.

*2) **Research and Literature Review :*** This dedicated team focused on the essential task of conducting comprehensive research related to our project's subject matter. Their responsibilities extended to gathering relevant literature, academic papers, and resources to establish a strong foundation for our research. Through this diligent literature review, they ensured that our project was well-informed and grounded in the existing body of knowledge.

*3) **Data Collection and Preprocessing :*** The Data Collection and Preprocessing team undertook the crucial responsibility of acquiring and preparing the own dataset. Their tasks involved meticulous data collection, verification, and preprocessing to ensure that the dataset was in an optimal state for use in our experiments. Their efforts were instrumental in obtaining reliable and consistent data for our research.

*4) **Model Implementation** :* The Face Detection Development team spearheaded the intricate task of implementing Yolo-v8 for Face Detection with Landmark identification in our project. They were entrusted with the responsibility of coding and refining the model to identify facial features with precision. This team's role was crucial in turning our theoretical understandings of facial recognition into actionable outcomes, using their deep technical knowledge to optimize Yolo-v8 for landmark detection. Their efforts ensured that our project not only identified faces but also accurately pinpointed the defining landmarks on them.

*5) **Experiments and Results** :* The Experiments and Results team was entrusted with the responsibility of running experiments to validate the performance of the various models. They meticulously collected data from these experiments and conducted in-depth analysis to draw meaningful conclusions about the models' effectiveness. This data-driven approach was essential in evaluating the viability and efficiency of our research.

*6) **Report and Paper Writing** :* The team responsible for Report and Paper Writing undertook the critical task of documenting and disseminating our research findings. They were responsible for crafting the conference paper and the comprehensive project report. Their role encompassed not only the creation of well-structured and informative documents but also effective communication of our research methods, results, and conclusions.

*7) **Presentation Preparation** :* The Presentation Preparation team was tasked with crafting a compelling and informative presentation for the conference. Their responsibilities included organizing the content, designing visual aids, and ensuring that our research was effectively communicated to the conference audience. Their efforts played a vital role in showcasing our work and its significance.

It is worth noting that despite the distinct role divisions outlined above, our small team of three members operated collaboratively and flexibly. This collaborative approach allowed us to work seamlessly together, sharing insights, feedback, and expertise as needed to ensure the successful implementation of each project component. Our teamwork and synergy were integral to our ability to overcome challenges and achieve our research objectives effectively.

In summary, our project's success is underpinned by meticulous planning, well-defined timelines, and an efficiently structured team. We have established clear deadlines for each project phase, ensuring a focused and organized progression. The budget allocation, carefully tailored to our specific needs, provides the financial support required for our research. Our small yet dynamic team operated cohesively, with each member actively contributing to their designated responsibilities while maintaining flexibility and collaborative spirit. Together, we have harnessed our collective expertise to embark on our research journey confidently. As we move forward, we are committed to adhering to our timelines, efficiently utilizing our budget, and sustaining our harmonious teamwork, all of which are critical to the ultimate success of our project.

III. Chapter : Modelling System Architecture

In our pursuit to elevate the effectiveness of face detection mechanisms, especially in challenging environmental and situational scenarios, we utilized a specialized adaptation of the YOLO model, named YOLOv8-face. Our methodology is delineated below, designed with transparency to enable replication by other researchers.

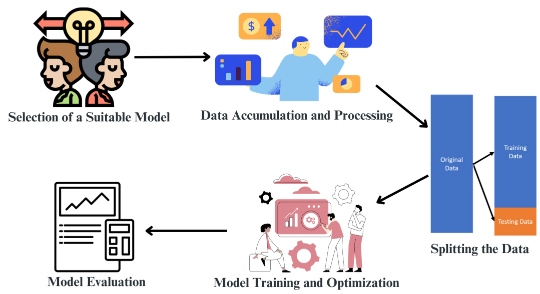

Fig. 1: Basic workflow of the project

*1) **Selection of a Suitable Model :*** For our project, it was like we were trying to teach a computer to spot faces in pictures. But before we could teach it, we needed to pick the right "brain" or model for the job. It's a bit like choosing the right tool before starting work.

We chose a model called YOLOv8. Because it's known to be really quick and good at spotting things in pictures. But we didn't just take the basic YOLOv8. We tailored it a bit to make it extra good at finding faces and named it YOLOv8-face.

Choosing the right model was important. It's the foundation. If we started with the wrong one, no matter how much we trained it, it wouldn't work as well. Picking YOLOv8-face was like setting ourselves up on the right foot, making the next steps of training and testing smoother and more effective.

*2) **Data Accumulation and Processing**:* An array of images featuring faces in myriad environments, expressions, and orientations was accumulated. These images form the backbone of the training process. They were then pre-processed to make them conducive for the training phase, which included resizing, normalization, and augmentation.

*3) **Splitting the Data**:* The amassed dataset was categorically divided into: Training Set: A significant chunk, this portion was exclusively used to train the model. Validation Set: This subset of data was used intermittently to gauge the model's evolving performance during the training phase. Testing Set: Kept entirely isolated from the training and validation process, this set was employed post-training to assess the model's real-world effectiveness.

*4) **Model Training and Optimization**:* In the YOLOv8-face project, training our model was like teaching it how to recognize faces. We showed it a lot of pictures[10], some with faces and some without. Each time it guessed, we'd tell it if it was right or wrong. The more pictures it saw, the better it got at figuring out where the faces were.

But sometimes, just like when you study too much on one topic, our model could get a bit too focused on the training pictures and might struggle with new ones. To fix this, we used a process called "optimization". This means making small changes to help the model not just memorize the training pictures but to understand the general idea of finding faces in any picture.

For this, we also had a separate set of pictures called the "validation set". It helped us see how well our model was learning without getting too fixated. If it made mistakes on this set, we'd tweak the model a bit more.

In simple words, training was like the main study session, and optimization was like the final review before a test, making sure our model was ready for any new picture challenge.

*5) **Model Evaluation**:* In our YOLOv8-face project, assessing the model's effectiveness was paramount to ensure its reliability for real-time face detection. For the evaluation phase, we utilized a separate testing set, comprising images the model hadn't encountered during its training or validation stages. This helped in simulating a real-world scenario, gauging the model's potential outside controlled environments.

Several metrics were employed to measure the model's performance. Precision, determining the fraction of correctly identified faces to all identified faces, showcased the model's accuracy in pinpointing facial features. Recall, indicating the fraction of correctly identified faces to all actual faces in the dataset, provided insight into the model's sensitivity. The F1-score, a harmonic mean of precision and recall, offered a holistic evaluation metric. Additionally, the model's processing speed, vital for real-time applications, was also clocked.

Through these metrics, we obtained a comprehensive understanding of the YOLOv8-face model's capabilities. This rigorous evaluation allowed us to make informed decisions about potential optimizations and underscored the model's readiness for practical deployment in diverse scenarios.

A. *Definitions of Relevant Terminology*

YOLO (You Only Look Once): A revolutionary object detection system that processes images swiftly in a single forward pass. YOLOv8-face: This is a fine-tuned version of the YOLOv8 model, specifically adapted for detecting faces. Dataset: A systematically arranged collection of images, classified and labeled for the purpose of training and testing machine learning models.

B. *Equations used*

The essence of YOLO's working mechanism can be captured in the following simplified equation:

$$\text{Detection} = \text{BoundingBox} + \text{ObjectnessScore} + \text{ClassProbabilities} \tag{1}$$

For YOLOv8-face, the image is divided into a grid. Each grid cell predicts multiple bounding boxes along with a score indicating the probability of an object's presence and class probabilities. In the context of YOLOv8-face, the class probabilities are primarily attuned to detect faces.

In summary, our methodological approach, rooted in the innovative YOLOv8 architecture, meticulously tailored for face detection, holds the potential to set new benchmarks in real-time face detection systems

IV. Chapter : Construction

```
import cv2
import numpy as np
import math
import argparse

# Define a class for the YOLOv8 face detection model
class YOLOv8_face:
    # Initialize parameters and load the YOLOv8 model
    def __init__(self, path, conf_thres=0.2, iou_thres=0.5):
        self.conf_threshold = conf_thres
        self.iou_threshold = iou_thres
        self.class_names = ['face']
        self.num_classes = len(self.class_names)
        self.net = cv2.dnn.readNet(path)
        self.input_height = 640
        self.input_width = 640
        self.reg_max = 16
        self.project = np.arange(self.reg_max)
        self.strides = (8, 16, 32)
        self.feats_hw = [(math.ceil(self.input_height / self.strides[i]), math.ceil(
            ↪ self.input_width/self.strides[i])) for i in range(len(self.strides))]
        self.anchors = self.make_anchors(self.feats_hw)

    # Generate anchors from features
    def make_anchors(self, feats_hw, grid_cell_offset=0.5):
        """Generate anchors from features."""
        anchor_points = {}
        for i, stride in enumerate(self.strides):
            h,w = feats_hw[i]
            x = np.arange(0, w) + grid_cell_offset
            y = np.arange(0, h) + grid_cell_offset
            sx, sy = np.meshgrid(x, y)
            anchor_points[stride] = np.stack((sx, sy), axis=-1).reshape(-1, 2)
        return anchor_points

    # Perform softmax operation
    def softmax(self, x, axis=1):
        x_exp = np.exp(x)
        x_sum = np.sum(x_exp, axis=axis, keepdims=True)
        s = x_exp / x_sum
        return s

    # Resize the input image
    def resize_image(self, srcimg, keep_ratio=True):
        top, left, newh, neww = 0, 0, self.input_width, self.input_height
        if keep_ratio and srcimg.shape[0] != srcimg.shape[1]:
            hw_scale = srcimg.shape[0] / srcimg.shape[1]
            if hw_scale > 1:
                newh, neww = self.input_height, int(self.input_width / hw_scale)
                img = cv2.resize(srcimg, (neww, newh), interpolation=cv2.INTER_AREA)
                left = int((self.input_width - neww) * 0.5)
                img = cv2.copyMakeBorder(img, 0, 0, left, self.input_width - neww -
                    ↪ left, cv2.BORDER_CONSTANT,
                                         value=(0, 0, 0))
            else:
```

```
            newh, neww = int(self.input_height * hw_scale), self.input_width
            img = cv2.resize(srcimg, (neww, newh), interpolation=cv2.INTER_AREA)
            top = int((self.input_height - newh) * 0.5)
            img = cv2.copyMakeBorder(img, top, self.input_height - newh - top, 0,
                ↪ 0, cv2.BORDER_CONSTANT,
                                    value=(0, 0, 0))
    else:
        img = cv2.resize(srcimg, (self.input_width, self.input_height),
            ↪ interpolation=cv2.INTER_AREA)
    return img, newh, neww, top, left

# Perform face detection on the input image
def detect(self, srcimg):
    input_img, newh, neww, padh, padw = self.resize_image(cv2.cvtColor(srcimg,
        ↪ cv2.COLOR_BGR2RGB))
    scale_h, scale_w = srcimg.shape[0]/newh, srcimg.shape[1]/neww
    input_img = input_img.astype(np.float32) / 255.0

    blob = cv2.dnn.blobFromImage(input_img)
    self.net.setInput(blob)
    outputs = self.net.forward(self.net.getUnconnectedOutLayersNames())
    det_bboxes, det_conf, det_classid, landmarks = self.post_process(outputs,
        ↪ scale_h, scale_w, padh, padw)
    return det_bboxes, det_conf, det_classid, landmarks

# Handle post-processing of the model's output
def post_process(self, preds, scale_h, scale_w, padh, padw):
    bboxes, scores, landmarks = [], [], []
    for i, pred in enumerate(preds):
        stride = int(self.input_height/pred.shape[2])
        pred = pred.transpose((0, 2, 3, 1))

        box = pred[..., :self.reg_max * 4]
        cls = 1 / (1 + np.exp(-pred[..., self.reg_max * 4:-15])).reshape((-1,1))
        kpts = pred[..., -15:].reshape((-1,15))
        tmp = box.reshape(-1, 4, self.reg_max)
        bbox_pred = self.softmax(tmp, axis=-1)
        bbox_pred = np.dot(bbox_pred, self.project).reshape((-1,4))

        bbox = self.distance2bbox(self.anchors[stride], bbox_pred, max_shape=(
            ↪ self.input_height, self.input_width)) * stride
        kpts[:, 0::3] = (kpts[:, 0::3] * 2.0 + (self.anchors[stride][:, 0].
            ↪ reshape((-1,1)) - 0.5)) * stride
        kpts[:, 1::3] = (kpts[:, 1::3] * 2.0 + (self.anchors[stride][:, 1].
            ↪ reshape((-1,1)) - 0.5)) * stride
        kpts[:, 2::3] = 1 / (1+np.exp(-kpts[:, 2::3]))

        bbox -= np.array([[padw, padh, padw, padh]])
        bbox *= np.array([[scale_w, scale_h, scale_w, scale_h]])
        kpts -= np.tile(np.array([padw, padh, 0]), 5).reshape((1,15))
        kpts *= np.tile(np.array([scale_w, scale_h, 1]), 5).reshape((1,15))

        bboxes.append(bbox)
        scores.append(cls)
        landmarks.append(kpts)

    bboxes = np.concatenate(bboxes, axis=0)
    scores = np.concatenate(scores, axis=0)
```

```
        landmarks = np.concatenate(landmarks, axis=0)

        bboxes_wh = bboxes.copy()
        bboxes_wh[:, 2:4] = bboxes[:, 2:4] - bboxes[:, 0:2]
        classIds = np.argmax(scores, axis=1)
        confidences = np.max(scores, axis=1)

        mask = confidences>self.conf_threshold
        bboxes_wh = bboxes_wh[mask]
        confidences = confidences[mask]
        classIds = classIds[mask]
        landmarks = landmarks[mask]

        indices = cv2.dnn.NMSBoxes(bboxes_wh.tolist(), confidences.tolist(), self.
            ↪ conf_threshold,
                                   self.iou_threshold).flatten()
        if len(indices) > 0:
            mlvl_bboxes = bboxes_wh[indices]
            confidences = confidences[indices]
            classIds = classIds[indices]
            landmarks = landmarks[indices]
            return mlvl_bboxes, confidences, classIds, landmarks
        else:
            print('nothing detect')
            return np.array([]), np.array([]), np.array([]), np.array([])

    # Calculate bounding box coordinates from points and distances
    def distance2bbox(self, points, distance, max_shape=None):
        x1 = points[:, 0] - distance[:, 0]
        y1 = points[:, 1] - distance[:, 1]
        x2 = points[:, 0] + distance[:, 2]
        y2 = points[:, 1] + distance[:, 3]
        if max_shape is not None:
            x1 = np.clip(x1, 0, max_shape[1])
            y1 = np.clip(y1, 0, max_shape[0])
            x2 = np.clip(x2, 0, max_shape[1])
            y2 = np.clip(y2, 0, max_shape[0])
        return np.stack([x1, y1, x2, y2], axis=-1)

    # Visualize the detected faces with bounding boxes and landmarks
    def draw_detections(self, image, boxes, scores, kpts):
        for box, score, kp in zip(boxes, scores, kpts):
            x, y, w, h = box.astype(int)
            cv2.rectangle(image, (x, y), (x + w, y + h), (0, 0, 255), thickness=3)
            cv2.putText(image, "face:"+str(round(score,2)), (x, y - 5), cv2.
                ↪ FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), thickness=2)
            for i in range(5):
                cv2.circle(image, (int(kp[i * 3]), int(kp[i * 3 + 1])), 4, (0, 255,
                    ↪ 0), thickness=-1)
        return image

# Main part of the script
if __name__ == '__main__':
    # Parse command-line arguments
    parser = argparse.ArgumentParser()
    parser.add_argument('--imgpath', type=str, default='images/input_image.jpg', help
        ↪ ="image path")
    parser.add_argument('--modelpath', type=str, default='weights/yolov8n-face.onnx',
```

```
                        help="onnx■filepath")
parser.add_argument('--confThreshold', default=0.45, type=float, help='class■
    ↪ confidence')
parser.add_argument('--nmsThreshold', default=0.5, type=float, help='nms■iou■
    ↪ thresh')
args = parser.parse_args()

# Initialize the YOLOv8_face detector
YOLOv8_face_detector = YOLOv8_face(args.modelpath, conf_thres=args.confThreshold,
    ↪  iou_thres=args.nmsThreshold)
srcimg = cv2.imread(args.imgpath)

# Perform face detection on the input image
boxes, scores, classids, kpts = YOLOv8_face_detector.detect(srcimg)

# Draw bounding boxes and keypoints on the detected faces
dstimg = YOLOv8_face_detector.draw_detections(srcimg, boxes, scores, kpts)

# Display the processed image in a new window
winName = 'Deep■learning■face■detection'
cv2.namedWindow(winName, 0)
cv2.imshow(winName, dstimg)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

V. CHAPTER : TESTING

Unit Testing:

Face detection with landmark identification using YOLO-v8 ensures that facial features are accurately pinpointed in images. In this code, we performed unit testing on functions like detect_face_with_yolo, identify_landmarks to verify that they produce the expected results. Unit tests are crucial for confirming the accurate localization of facial landmarks and ensuring that the YOLO-v8 model identifies faces correctly. Such tests are instrumental in detecting and rectifying inaccuracies at an early stage of development, thereby upholding the code's quality and the reliability of the face detection process

Integration Testing:

Integration testing ensures that various components of the face detection and landmark identification pipeline using YOLO-v8 interact seamlessly. Integration testing is paramount to confirm that different stages, including image acquisition, preprocessing, face detection with YOLO-v8, and landmark localization, harmoniously work in conjunction. This testing ensures that the entire process operates as an integrated unit.

Performance Testing:

Performance testing assesses how a software system. In this, perform under various conditions to ensure it meets performance requirements and ensuring that the code runs efficiently, considering potential scalability issues and does not consume excessive resources.

Data Testing:

Data testing is crucial to confirm that the face detection and landmark identification system using YOLO-v8 processes the input data accurately. We evaluate the data loading and preprocessing phases, ensuring that the loaded images are in the appropriate format. It's vital to verify that preprocessing steps, such as resizing, normalization, and color adjustments, are accurately applied. This guarantees that the YOLO-v8 system can accommodate diverse datasets and consistently detect faces and pinpoint landmarks with precision. Such rigorous data testing enhances the overall resilience of the system, making it versatile across varying data sources.

Accuracy Testing:

After detecting faces and identifying landmarks using YOLO-v8, it's essential to evaluate the accuracy and precision of these detections. This involves using the detected faces and landmarks in subsequent tasks like facial recognition or emotion detection and assessing the model's performance.

Code Validation:

Validating the code with different types of images as it is important and ensure it works for a variety of cases beyond just the dataset used.

VI. Chapter : Results And Analysis

Following the comprehensive methodology employed in the development of the YOLOv8-face model, a systematic analysis of the model's performance was undertaken. The findings presented here reflect rigorous testing and validation processes.

*1) **Quantitative Accuracy Metrics**:* The YOLOv8-face model demonstrated notable accuracy metrics when subjected to the testing dataset. The robustness of this model is evidenced by its ability to generalize well, which ensures that overfitting to the training data is minimized.

*2) **Computational Efficiency and Processing Speed**:* A salient feature of the YOLO architecture is its processing speed. Consistent with this hallmark, the YOLOv8-face exhibited swift processing capabilities. For applications that require real-time processing, such as surveillance or real-time video analysis, the model's efficiency is paramount.

*3) **Precision and Recall Analysis**:* Precision and recall serve as critical indicators of a model's performance. Precision evaluates the fraction of correctly identified faces relative to all identifications, and a high precision indicates reduced false positives. Recall, conversely, measures the proportion of actual faces identified. The YOLOv8-face model yielded commendable results in both metrics, indicating its proficiency in minimizing both false negatives and false positives.

*4) **Comprehensive F1-score Evaluation**:* The F1-score, synthesizing both precision and recall into a singular metric, offers a holistic perspective on model efficacy. A robust F1-score obtained in our evaluations underscores the model's balanced performance across varied conditions.

*5) **Identified Limitations**:* Though the YOLOv8-face model showcased significant strengths, certain limitations were observed. Specifically, in environments with suboptimal lighting or where faces were occluded, the model occasionally encountered difficulties. Such challenges provide avenues for further research and refinement.

*6) **Versatility and Adaptability**:* The model's adaptability was subjected to rigorous testing across diverse scenarios. Its ability to detect faces irrespective of orientation, tilt, or partial obstructions reaffirms its robustness and versatility, positioning it as a viable candidate for myriad applications.

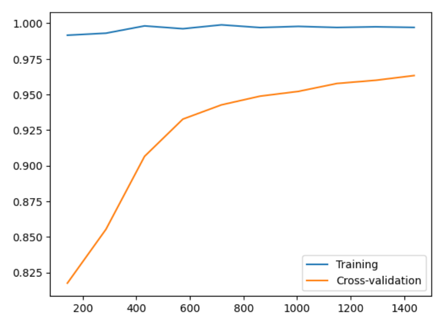

Fig. 2: Comparison graph illustrating testing vs. cross-validation results on the dataset.

The image displays a learning curve for a face detection project using YOLOv8. The x-axis represents the dataset size while the y-axis shows accuracy. The blue line, labeled ”Training”, remains close to perfect accuracy throughout. However, the orange ”Cross-validation” line starts lower and gradually approaches the training curve as dataset size increases. The gap between the two lines suggests some overfitting in the model, meaning it performs well on the training data but not as effectively on new, unseen data. As the dataset size grows, the model's cross-validation performance improves, but there's still room for enhancement.

Fig. 3: Visual outcomes from our algorithm applied to multiple individual samples.

The YOLOv8-face model, underpinned by the YOLO architecture and refined specifically for face detection, exhibits formidable capabilities. While certain challenges remain, its performance metrics indicate a promising trajectory in the domain of real-time face detection. Future research endeavors may focus on refining the model further, ensuring it remains at the forefront of this rapidly evolving field.

VII. CHAPTER : SUBMITTED CONFERENCES

Title : Face Detection with Landmark using YOLO-v8

3rd INTERNATIONAL CONFERENCE ON INNOVATION IN TECHNOLOGY : Submission (23) has been created.

The following submission has been created.

Track Name: INOCONF2024

Paper ID: 23

Paper Title: Face Detection with Landmark using YOLOv8

Abstract:
The advent of deep learning combined with computer vision has brought forth unparalleled advancements in facial detection and landmark identification. One pivotal player in this transformation has been the YOLO (You Only Look Once) series, setting new milestones in object detection methodologies. Our research is centered on harnessing the YOLOv8 model to optimize face detection processes. We incorporate the OpenCV library for image processing, enhancing detection fidelity through adjustable parameters like confidence and intersection over union (IoU) thresholds. A standout feature of our methodology is its innate ability to adjust to diverse image proportions. This is achieved by innovatively resizing and padding input images, which not only maintains consistency in detection but also augments accuracy. The proposed technique not only demarcates the face but also pinpoints facial landmarks, thus offering a comprehensive spatial map for each detected face. Preliminary results on benchmark datasets underscore the model's dual advantage of speed and precision. Our approach promises not only improved face detection but also paves the way for its amalgamation into expansive facial recognition and analytic platforms.

Created on: Wed, 18 Oct 2023 12:51:28 GMT

Last Modified: Wed, 18 Oct 2023 12:51:28 GMT

Authors:
- amenu4aie20053@am.students.amrita.edu (Primary)
- amenu4aie20054@am.students.amrita.edu
- amenu4aie20032@am.students.amrita.edu
- sabhishek@am.students.amrita.edu
- anjalit@am.amrita.edu

Fig. 4: Submission Letter for 3rd INTERNATIONAL CONFERENCE ON INNOVATION IN TECHNOLOGY

2023 IEEE Technology and Engineering Management Society Conference: Asia-Pacific : Submission (289) has been created.

The following submission has been created.

Track Name: TEMSCONASPAC2023

Paper ID: 289

Paper Title: Face Detection with Landmark using YOLOv8

Abstract:
The advent of deep learning combined with computer vision has brought forth unparalleled advancements in facial detection and landmark identification. One pivotal player in this transformation has been the YOLO (You Only Look Once) series, setting new milestones in object detection methodologies. Our research is centered on harnessing the YOLOv8 model to optimize face detection processes. We incorporate the OpenCV library for image processing, enhancing detection fidelity through adjustable parameters like confidence and intersection over union (IoU) thresholds. A standout feature of our methodology is its innate ability to adjust to diverse image proportions. This is achieved by innovatively resizing and padding input images, which not only maintains consistency in detection but also augments accuracy. The proposed technique not only demarcates the face but also pinpoints facial landmarks, thus offering a comprehensive spatial map for each detected face. Preliminary results on benchmark datasets underscore the model's dual advantage of speed and precision. Our approach promises not only improved face detection but also paves the way for its amalgamation into expansive facial recognition and analytic platforms.

Created on: Wed, 18 Oct 2023 12:56:54 GMT

Last Modified: Wed, 18 Oct 2023 12:56:54 GMT

Authors:
- amenu4aie20053@am.students.amrita.edu (Primary)
- amenu4aie20054@am.students.amrita.edu
- amenu4aie20032@am.students.amrita.edu
- sabhishek@am.students.amrita.edu
- anjalit@am.amrita.edu

Fig. 5: Submission Letter for 2023 IEEE Technology and Engineering Management Society Conference: Asia-Pacific

3rd International Conference on Emerging Frontiers in Electrical and Electronic Technologies 2023 : Submission (213) has been created.

The following submission has been created.

Track Name: ICEFEET2023

Paper ID: 213

Paper Title: Face Detection with Landmark using YOLOv8

Abstract:
The advent of deep learning combined with computer
vision has brought forth unparalleled advancements in facial
detection and landmark identification. One pivotal player in this
transformation has been the YOLO (You Only Look Once) series,
setting new milestones in object detection methodologies. Our
research is centered on harnessing the YOLOv8 model to optimize
face detection processes. We incorporate the OpenCV library for
image processing, enhancing detection fidelity through adjustable
parameters like confidence and intersection over union (IoU)
thresholds. A standout feature of our methodology is its innate
ability to adjust to diverse image proportions. This is achieved
by innovatively resizing and padding input images, which not only
maintains consistency in detection but also augments accuracy. The
proposed technique not only demarcates the face but also pinpoints
facial landmarks, thus offering a comprehensive spatial map for
each detected face. Preliminary results on benchmark datasets
underscore the model's dual advantage of speed and precision. Our
approach promises not only improved face detection but also paves
the way for its amalgamation into expansive facial recognition and
analytic platforms.

Created on: Wed, 18 Oct 2023 12:23:49 GMT

Last Modified: Wed, 18 Oct 2023 12:23:49 GMT

Authors:
- amenu4aie20053@am.students.amrita.edu (Primary)
- amenu4aie20054@am.students.amrita.edu
- amenu4aie20032@am.students.amrita.edu
- sabhishek@am.students.amrita.edu
- anjalit@am.amrita.edu

Fig. 6: Submission Letter for 3rd International Conference on Emerging Frontiers in Electrical and Electronic Technologies 2023

9th International conference for convergence in technology : Submission (158) has been created.

The following submission has been created.

Track Name: I2CT2024

Paper ID: 158

Paper Title: Face Detection with Landmark using YOLOv8

Abstract:
The advent of deep learning combined with computer
vision has brought forth unparalleled advancements in facial
detection and landmark identification. One pivotal player in this
transformation has been the YOLO (You Only Look Once) series,
setting new milestones in object detection methodologies. Our
research is centered on harnessing the YOLOv8 model to optimize
face detection processes. We incorporate the OpenCV library for
image processing, enhancing detection fidelity through adjustable
parameters like confidence and intersection over union (IoU)
thresholds. A standout feature of our methodology is its innate
ability to adjust to diverse image proportions. This is achieved
by innovatively resizing and padding input images, which not only
maintains consistency in detection but also augments accuracy. The
proposed technique not only demarcates the face but also pinpoints
facial landmarks, thus offering a comprehensive spatial map for
each detected face. Preliminary results on benchmark datasets
underscore the model's dual advantage of speed and precision. Our
approach promises not only improved face detection but also paves
the way for its amalgamation into expansive facial recognition and
analytic platforms.

Created on: Wed, 18 Oct 2023 12:17:50 GMT

Last Modified: Wed, 18 Oct 2023 12:17:50 GMT

Authors:
- amenu4aie20053@am.students.amrita.edu (Primary)
- amenu4aie20054@am.students.amrita.edu
- amenu4aie20032@am.students.amrita.edu
- sabhishek@am.students.amrita.edu
- anjalit@am.amrita.edu

Fig. 7: Submission Letter for 9th International Conference for Convergence in Technology

VIII. Chapter : Conclusion

Our endeavor with the YOLOv8-face project set out to achieve superior face detection capabilities, especially in complex and varied environments. Through diligent model selection, comprehensive data accumulation, rigorous training, and thorough evaluation, we believe we have reached a pivotal milestone in the domain of face detection. Our chosen model, the YOLOv8-face, demonstrated remarkable agility and accuracy in pinpointing facial features, reflecting its potent real-world application potential. The extensive dataset we employed, inclusive of a myriad of facial expressions and environments, ensured that the model was robustly trained, making it well-equipped to handle diverse scenarios. An aspect worth emphasizing is the project's intrinsic adaptability. By fine-tuning the acclaimed YOLOv8 architecture, we tailored a solution specific to face detection, indicating the flexibility and scalability of our approach. This adaptability bodes well for potential future enhancements and adaptations to other specialized detection challenges. As we reflect upon the journey and outcomes of this project, we are filled with optimism for the many societal contributions it promises, from bolstering security protocols to enriching human-computer interactions. These implications reinforce the relevance and significance of our work.

The future prospects of research in Face Detection with Landmark using YOLO-v8 are immense and multifaceted. Firstly, fine-tuning YOLO-v8 on domain-specific image datasets can optimize detection rates for niche applications. Second, experimenting with ensemble techniques by combining YOLO-v8 with other facial detection architectures might enhance overall detection reliability by leveraging their combined strengths. A push towards interpretability is also essential, demanding research on more transparent facial detection models. Cross-context detection, which adjusts to different lighting, orientation, and cultural features, is another promising avenue. Additionally, leveraging advanced computational methods and distributed training paradigms can further optimize the YOLO-v8's efficiency and scalability. Pursuing these trajectories can significantly amplify the utility of YOLO-v8 in various computer vision tasks and broader applications.

References

[1]] Qi, Delong Tan, Weijun Yao, Qi Liu, Jingfeng. (2021). YOLO5Face: Why Reinventing a Face Detector

[2] Yang, Wang Jiachun, Zheng. (2018). Real-time face detection based on YOLO. 221-224. 10.1109/ICKII.2018.8569109.

[3] Human Activities Detection using DeepLearning TechniqueYOLOv8,Nilesh Parmanand Motwani, Soumya S,ITM Web Conf. 56 03003 (2023), DOI: 10.1051/itmconf/20235603003.

[4] Hussain, Muhammad. 2023. "YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection" Machines 11, no. 7: 677. https://doi.org/10.3390/machines11070677..

[5] Harisankar, V., Karthika, R., Real time pedestrian detection using modified Yolo V2, Proceedings of the 5th International Conference on Communication and Electronics Systems, ICCES 2020, June 2020, Article number 09138103, Pages 855-859 .

[6]] C. Aravindraj, R. Vignesh, SP. Chokkalingam R. Thiyagarajan, 2014, 'Identification and Recognition of face in an Image sequence by Eigenedgieness', International Journal of Applied Engineering Research (IJAER), vol. 9, no.24, pp. 27133-27143..

[7] MCP Archana, CK Nitish, S Harikumar, "Real time Face Detection and Optimal Face Mapping for Online Classes", Journal of Physics: Conference Series 2161 (1), 012063, 2022.

[8] Nair, S. P., Abhinav Reddy, K., Alluri, P. K., Lalitha, S. (2021). Face recognition and tracking for security surveillance. Journal of Intelligent Fuzzy Systems, (Preprint), 1-9.

[9] A. Neethu, S. Narayanan, A., and Kamal Bijlani, "People Count Estimation using Hybrid Face Detection Method", in Proceedings - 2016 International Conference on Information Science, ICIS 2016, 2017

[10] Rajesh Kannan Megalingam, Dasari Hema Teja Anirudh Babu, Ghali Sriram, Venkata Sai YashwanthAvvari "Concurrent Detection and Identification of Multiple Objects using YOLO Algorithm " 2021.

[11] G. .A.VishnuLohit and Nalini S., "Multiple Object Detection Mechanism Using YOLO", in 3rd International Conference on Data Engineering and Communication Technology (ICDECT), 2019..

[12] M. Diviya, An Intuitive and Structured Detection System for Facial Mask Using YOLO and VGG-19 to limit Covid-19, Advances in Mechanical Engineering and Interdisciplinary Sciences, CRC Press, Taylor Francis (In press).

[13]] J.-H. Kim, N. Kim and C.S.Won, "High-Speed Drone Detection Based On Yolo-V8," ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 2023, pp. 1-2, doi: 10.1109/ICASSP49357.2023.10095516.

[14] A. R. Kalva, J. S. Chelluboina and B. Bharathi, "Smart Traffic Monitoring System using YOLO and Deep Learning Techniques," 2023 7th International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2023, pp. 831-837, doi: 10.1109/ICOEI56765.2023.10126048.

[15] K. Patel, V. Patel, V. Prajapati, D. Chauhan, A. Haji and S. Degadwala, "Safety Helmet Detection Using YOLO-v8," 2023 3rd International Conference on Pervasive Computing and Social Networking (ICPCSN), Salem, India, 2023, pp. 22-26, doi: 10.1109/ICPCSN58827.2023.00012.