Cheating Detection Through CCTV using YOLOv7

Mizanul Ridho Aohana[1[0009-0006-2228-6718]], Fitri Bimantoro[1[0000-0003-1813-0279]], I Gede Pasek Suta Wijaya[1[0000-0002-3813-013X]]

[1] Dept Informatics Engineering, University of Mataram, Jl. Majapahit 62, Mataram, Lombok NTB, Indonesia
bimo@unram.ac.id

Abstract. In the current era, digitalization has become essential for most educational institutions to measure students' knowledge and abilities through examinations. However, students often find various ways to cheat, such as exchanging answer sheets, cheating friends, using hidden notes, or providing answer codes. Students often ignore integrity, which is a crucial factor in the exams. Human control is often inconsistent and limited by focus. Therefore, this research proposes a solution for solving the cheating problem using computer vision. The focus of this research is monitoring the suspicious behavior of students (physical) during the examination through CCTV cameras. The method used to solve the problem is the YOLO (You Only Look Once) algorithm. Subsequently, a comparative analysis is conducted among three versions of the YOLO algorithm: YOLOv5, YOLOv6, and YOLOv7. In this study, the accuracy results for each algorithm variation are 43%, 37%, and 51%, respectively. The results show that the YOLO algorithm can detect actions of cheating in a classroom environment. The existence of imbalanced classes in the dataset is the main factor that affects the performance value, especially in the Giving Code and Exchange Paper classes. Consequently, an extended experiment is undertaken by adding 10% to the dataset for the imbalanced classes. With this augmentation, the highest accuracy recorded is 60%, given by YOLOv7, reflecting a noteworthy 9% increase in accuracy.

Keywords: Cheating Detection, Physical Exam, Deep Learning, YOLOv5, YOLOv6, YOLOv7

1 Introduction

Technology has been proven to optimize the learning process [1] and has been shown to improve the quality of student understanding [2]. In addition, teaching staff can also improve the learning experience provided with the help of technology [3]. An examination is one of the essential aspects in educational institutions to measure how far the ability and knowledge of students related to the material that has been given [4]. Written exams [5] are among the most widely used methods [6]. In this modern era, there are many ways to automate traditional exams [7], including technological assistance [8].

The main thing that needs to be considered during the exam is the integrity of students in taking the exam [9]. In more specific terms, academic integrity is described as “a dedication to six core values: honesty, trust, fairness, respect, responsibility, and courage... these fundamental values serve to guide and enhance ethical decision-making abilities and behavior." [10]. Integrity is influenced by various factors [11] that can affect personality, especially the habituation of the learning environment at home [12].

However, in the implementation of the exam, there are often indications of cheating that disrupt the level of academic integrity [13]. Cheating is a dishonest behavior exhibited by students when completing assignments or examinations [14]. In most cases, this behavior is typically driven by a low sense of self-efficacy [15]. Cheating can be done using various methods [16], which aim to increase exam scores [17] and avoid failing the exam [18]. This is undoubtedly very dangerous in educational institutions because it affects the quality of student’s outcome [19].

Computer vision in various fields has been widely used, including education [20]. Installing Closed Circuit Television (CCTV) in classrooms can also facilitate the application of computer vision algorithms [21]. One of the popular algorithms in computer vision today is YOLO (You Only Look Once) [22], [23]. In its development, YOLO is often used to detect and recognize objects such as mask detection and social distancing [24], clothing attraction detection [25], traffic accidents [26] and counting the number of vehicles [27]. Based on some information from the references above, this algorithm is also very possible to detect student cheating when taking an exam [21].

The YOLO algorithm has various variants that can be used to detect various objects [28]. The emergence of many variants makes it an exciting choice to test individually. The resulting performance will, of course, vary depending on the algorithm's architecture, the dataset's quality, and the dataset's complexity [29].

Thus, this research will examine detailly some YOLO algorithms and compare their performance in identifying student cheating during written exams using CCTV system monitoring. The overall research aims to optimize cheating detection in the exam environment and contribute to developing exam security solutions based on computer vision technology.

2 Related Work

Examinations have an important role in measuring a student's understanding and reasoning, and they also play a crucial role in measuring the success of the learning process in educational institutions. Cheating during exams has become a classic and significant problem for educational institutions. During physical exams, students often use unfair means to pass the exam. Students are always looking for ways to cheat, mainly due to the negligence of invigilators.

Frequently used exam proctoring methods usually rely on the consistency of human proctors, which can be tedious, time-consuming, labor-intensive, and ineffective in preventing students from cheating. Therefore, educational institutions need to streamline these problems by implementing computer vision technology so that supervision can be done automatically.

In recent years, various studies have been conducted to create technologies that can overcome the solution to these problems. The use of deep learning algorithms has proven effectively for detection and classification problems. Previous research [30] has made an intelligent exam supervisor system that can detect student cheating based on the movement of examinees. The methods used are Random Forest, Logistic Reg, LSTM, 2DCNN, CNN-LSTM, Conv-LSTM, and CNN-BiGRU. From the comparison of these methods, the highest accuracy was obtained at 97.7% in the Random Forest and LSTM methods.

Previous research [31], developed an intelligent monitoring system that uses image recognition monitored live on streaming video from the exam room. The system divides the video frame into predefined candidate seat areas and alerts the invigilator if the limits are exceeded.

In previous research, a model was developed using spatial-temporal features to detect abnormal behaviour from students [32]. The proposed model can identify movements such as turning around and raising hands with an accuracy rate of 93.3%.

CCTV modelling has also detected various suspicious activities during exams [33]. The model recognizes activities such as hand contact and stealing classmates' exam sheets based on a certain threshold. That project converts video input from a camera (CCTV) into image frames. That study uses a Gaussian Filter to remove background, foreground, and noise for pre-processing. Suspicious activity detection is performed using the Haar-like feature algorithm developed by Viola and Jones. The proposed model successfully identified heads with 70% accuracy, hands with 72% accuracy, and faces with 84% accuracy.

Research [34] discusses using cameras for efficient classroom monitoring using motion detection algorithms and image processing. The proposed idea is an alternative solution categorizing body movements into legal and illegal activities. That research uses deep learning InceptionV3 and gets an error of less than 10%.

Using the concept of deep learning [35], a cheating detection model was successfully created during online exams using the Deep CNN method. Algorithm implementation was carried out on the front and rear cameras with accuracy achieved 99.83% and 99.81%, respectively.

Another study using CCTV [21] tried to automatically detect abnormal or cheating activities in the exam using the Open Pose method. The model was developed by detecting the body posture of students during the exam using CCTV. If the number of cases detected exceeds a predetermined threshold, it will be detected as cheating, and a report will be sent to the examiner. The accuracy obtained in that study was 63%.

Trying to dig more deeply [36], there is research using BRISK, HOG, MSER, SURF and SURF&HOG feature extraction to predict five classes of student cheating actions in the classroom. That research resulted in 91% accuracy.

The research discusses the utilization of computer vision methods through CCTV to detect anomalous activities of students during exams. That research centres on monitoring students' suspicious behaviour during physical exams through CCTV cameras. In making the model, that research used YOLOv3 and got an accuracy of 88.03% [37].

Research on an electronic cheating detection model that allows supervisors to perform Student/Person Detection and Tracking, Detect suspicious activity, Create alerts, and Mark attendance. Students are detected from videos using YOLOv7, and DeepSort trackers are used to track people detected by the YOLOv7 algorithm. The accuracy obtained in that study is 81.67% [38].

Based on some of the references described previously, that research will focus on using the YOLO algorithm, as it has been proven to have a pretty good level of accuracy in predicting student activity. The selection of that algorithm is also based on its significant level of existence today in the domain of computer vision problems, especially in the context of object detection and object classification.

3 Research Methods

3.1 Proposed Work

Deep learning has many important factors that can affect its success, so it is essential to do careful planning [39]. Therefore, before starting this research, it is necessary to design a comprehensive process so that the steps can be carried out in a structured manner to achieve optimal results. In the context of this research, there is a model development flow that starts from data collection to model evaluation. The following is a series of stages that we apply.

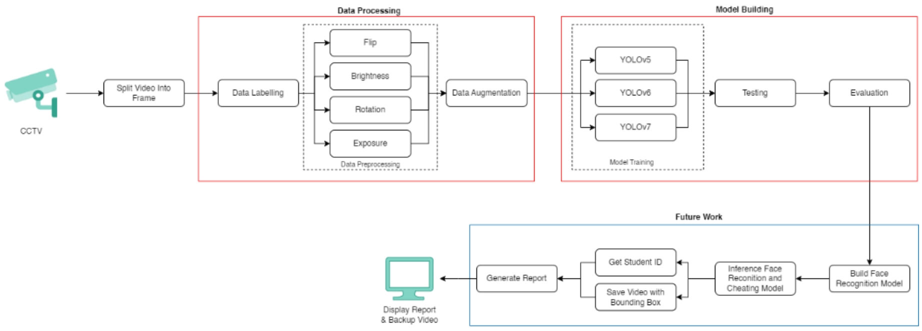

Fig. 1. Proposed Work Plan

Based on Fig. 1, the model development starts with data collection from CCTV and then proceeds with the separation of video into photo frames. The data obtained will be processed through the data preprocessing stage before modelling. After the model is successfully created, the following process includes a testing and evaluation stage to measure the model's performance. At this stage, the model will be tested using data not involved in the training process to ensure its reliability and effectiveness. The results of this evaluation will help refine the model, if necessary, before full implementation. Finally, after the testing and evaluation process is complete, the best model for recognizing problems will be obtained.

In future work, there is potential to develop the system further using face recognition. Through this model, the system can automatically identify students suspected of cheating during exams. Every suspicious gesture or action will be meticulously recorded and stored as part of the exam report. The recorded outcomes will be integrated into the exam report data and presented to the respective course instructors. Consequently, the utilization of facial recognition technology not only enhances security during exams but also provides valuable information to instructors for further evaluation. This development can be a positive step towards creating a fair and trustworthy exam environment, supporting effective exam management, and offering innovative solutions to potential cheating activities.

3.2 Data Collection

The dataset in this research was taken using CCTV from Hikvision (DS-2CD1121-I) with a resolution of 1920x1080, 25 fps in room A3-01, located in Building A, Informatics Engineering Dept., Faculty of Engineering, University of Mataram. The room has a capacity of 35 people and has an area of about 40 m^3.

This dataset contains 17 students who simulate an exam and demonstrate five conditions: normal, cheating, exchanging paper, giving code, and looking for friends. Dataset collection is carried out in about an hour and 30 minutes. The following is an example of a sample dataset.

Fig. 2. Dataset Collecting

Fig. 2 shows that the dataset obtained produces clear enough images for the model training process. However, the dataset is taken using CCTV in video format, the dataset needs to convert videos into frames or images before further processing. This extraction process is done using the Python language with the help of the OpenCV library, where frames will be taken every 7 seconds.

3.3 Data Pre-Processing

From the dataset that has been collected, it is necessary to do data preprocessing, such as removing unclear images and increasing the brightness of the image. In this case, the author uses the Roboflow platform to annotate the dataset to separate one label from another. Fig. 3 is an example of the annotation result.

Fig. 3. Dataset Annotation

Fig. 3, a dataset has undergone an annotation process. This annotation process involves grouping the data into five different classes, namely Cheat Sheet, Exchange Paper, Giving Code, Looking Friend and Normal. These data were thoroughly annotated, starting from the examinee's head to hands, to show the body gestures the examinee exhibited clearly.

Once the process of annotating the dataset was complete, the authors proceeded to apply several preprocessing steps to the data. This process involved operations such as horizontal flip, rotation (between -5° to 5°), brightness enhancement (between -25% to 25%), and exposure enhancement (between -10% to 10%) [40] [41]. In addition, the dataset was augmented three times to improve the performance of the model in object detection [42], bringing the total amount of annotated data to 10,252 [43]. This data is divided into various classes, with the following distribution: Normal class has 3,975 annotations, Looking Friend has 3,897 annotations, Cheat Sheet has 1,890 annotations, Giving Code has 372 annotations, and the last class, Exchange Paper, has 136 annotations.

3.4 Object Detection

Real-time object detection is essential in computer vision systems [38]. Object detection involves classifying and locating potential objects in an image [44]. Many algorithms can be used in object detection, such as Faster R-CNN, SSD, RetinaNet, and YOLO [38].

This research will focus on using the YOLO algorithm to build a cheating detection model. There are several variants of the YOLO algorithm that will be used in the model series, namely YOLOv5s, YOLOv5m, YOLOv5l, YOLOv6s, YOLOv6m, YOLOv6l, and YOLOv7. It is important to note that Ultralytics developed YOLOv5 and YOLOv7, while YOLOv6 was developed by Meituan [45] .

In general, the YOLO architecture can be described as an object detection approach that processes the image and generates object predictions in a single feedforward pass through the neural network. This approach enables fast and efficient object detection. Using various variants of YOLO, we will evaluate the performance and accuracy of each model in detecting suspicious behaviour during exams. The following is the architecture of YOLO [46].

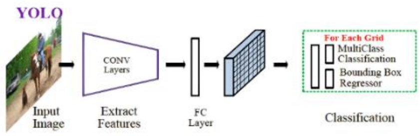

Fig. 4. YOLO Architecture [47]

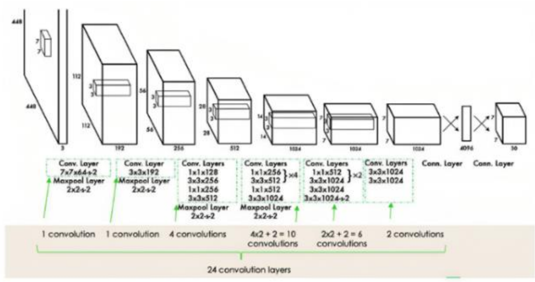

Fig. 5. YOLO Layers [46]

In Fig. 4 and Fig. 5, the YOLO algorithm starts with image input as the first step. This image then goes through a series of feature extraction stages in a convolutional neural network (CNN). This stage aims to identify unique features present in the image, such as object edges, textures, colours, and other patterns. This feature extraction stage results in a high-dimensional data representation that describes the image more abstractly.

Once the features have been successfully extracted, the next step is sending them to the FC (Fully Connected) layer in the neural network. This FC layer serves to combine and process the previously extracted features. Here, the processing involves more complex calculations and the formation of more abstract representations.

The output of the FC layer will be the final determinant in the classification process in the YOLO algorithm. This output includes information about the location of the object (bounding box coordinates), the detected object's class, and the detection's confidence level. In other words, the YOLO algorithm uses the features that have been processed through the FC layer to produce complete information about the objects in the image, including what the object is, where it is located, and the extent of its confidence.

With this approach, YOLO can efficiently and accurately detect objects in images by combining advanced feature extraction with information processing in the FC layer to provide robust classification results.

3.5 Object Tracking and Cropping

Object tracking is one of the fundamental aspects of computer vision that tries to detect and track every object in a series of images [48]. In its implementation, the tracking target will be determined in the first frame and must be detected and tracked in subsequent video frames.

In this research, the process of tracking and cutting objects is done using a web-based tool called Roboflow. This tool plays an important role in labelling objects, and the Roboflow output dataset is divided into three main parts, namely training data (train), validation data (validation), and test data (test). Each dataset has two main contents: a text file (txt) containing the label annotation coordinates and the corresponding image file (image).

3.6 Metrics Evaluation

Confusion Matrix is one of the most frequently used evaluation methods in machine learning, especially supervised learning [49]. The confusion matrix contains four evaluation values, namely True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) [50].

Table 1. Confusion Matrix Theory

	Prediction		
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

Based on Table 1, the evaluation matrix can be calculated in accuracy, precision, and recall. The following equation is used to calculate the evaluation value [51].

$$Acc = \frac{TP+TN}{TP+FN+FP+TN} \tag{1}$$

$$P = \frac{TP}{TP+FP} \tag{2}$$

$$R = \frac{TP}{TP+FN} \tag{3}$$

Accuracy (Acc) is one of the testing methods based on the overall level of closeness between the predicted value and the actual value. Precision (P) is the accuracy of predictions with correct labels with positive results. At the same time, Recall (R) represents the quantity of quality in the entire prediction process [52].

In the YOLO algorithm, one of the leading indicators to assess model performance is calculating the mAP (Mean Average Precision) value [53]. The mAP metric is a commonly used metric in object detection tasks to measure the extent to which the model can detect and classify objects with high accuracy [54].

Conceptually, mAP will calculate the average precision for each object class [55]. Precision measures the extent to which the detection made by the model is the intended object, and it is calculated by counting the number of true positives (TP) divided by the total number of positive predictions made by the model [56].

$$AP = \sum_{i=1}^{N} P(i)\Delta \mathrm{R(i)} \tag{4}$$

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP(img_i) \tag{5}$$

Where AP defines the average precision, N defines the number of classes, and img_i represents the i-th image iteration. The AP and mAP values range from 0-1, while the ideal value for both benchmarks is 1.

4 Experiments and Results

4.1 Result and Discussion

In this research, experiments have been conducted using Python programming language using pyTorch on an NVIDIA V100 device (Google Collab). The model has been trained with a learning rate of 0.005, batch size 16, 100 epoch, and image size 640x640 [57][58][59], [60]. This time, the dataset division in training is divided into 70% of the total dataset, with 20% of the data for validation and 10% for testing [61], This is also based on research conducted previously [62] and uses a similar composition in dealing with unbalanced datasets, 70% of the data is used to train, 10% for validation and 20% for testing.

The algorithms to be tested are YOLOv5s, YOLOv5m, YOLOv5l, YOLOv6s, YOLOv6m, YOLOv6l, and YOLOv7. Please note that in the YOLOv7 algorithm, researchers cannot try to use the medium and large versions of the YOLO model due to resource limitations.

In the experiments, some datasets have imbalances in 2 classes, namely the Exchange Paper and Giving Code classes. The dataset details can be seen in Fig. 6.

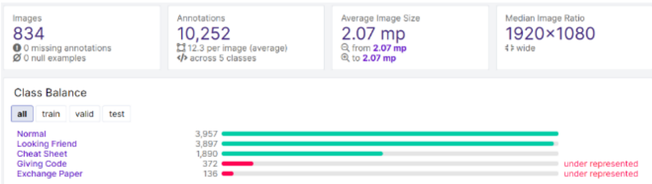

Fig. 6. Dataset

The Imbalanced dataset as shown in Fig. 6, the model will be forced to provide the best performance with a disproportionate dataset. Furthermore, regarding the results of the analysis that has been done, it is found that each algorithm has varying performance in modelling with the same dataset.

4.2 Comparative Analysis

The model summary can be seen in Table 2 based on the model training results.

Table 2. Model Result

Algorithm	Indicators			
	Param.	GFLOPs	mAP:50	mAP50:95
YOLOv7	36,50 M	103.2	0,559	0,344
YOLOv6l	59,54 M	150.51	0,535	0,333
YOLOv6m	34,81 M	85.63	0,505	0,311
YOLOv6s	18,50 M	45.17	0,504	0,305
YOLOv5l	46,13 M	107.7	0,555	0,363
YOLOv5m	20,87 M	47.9	0,497	0,337
YOLOv5s	7,02 M	45,15	0,549	0,351

From Table 2, it can be seen that YOLOv7 has a fairly low GFLOPs rate and a relatively high mAP in both metrics (mAP:50 and mAP 50:95). This indicates that the model performs well in detecting objects with fairly good accuracy over a wide variety of object sizes.

YOLOv6l has a higher GFLOPs rate than YOLOv7, but its mAP is slightly lower. This may indicate that this model requires more computation power for similar performance to YOLOv7.

Meanwhile, the YOLOv6m has a lower GFLOPs rate than larger models (such as the YOLOv6l) but has a lower mAP. Opting for YOLOv6m could be advantageous when faced with computational power limitations. YOLOv6s, a more lightweight model with significantly lower GFLOPs, entails a trade-off with detection performance compared to larger models. This model may be a suitable choice when prioritizing faster execution with minimal compromise on accuracy.

YOLOv5l has a moderate level of GFLOPs and a relatively high mAP. It may be a good choice if looking for a balance between performance and computation power. YOLOv5m is a lighter model, which can run faster but has a lower mAP than the larger model. YOLOv5s is the lightest model in terms of GFLOPs but still has a competitive mAP.

The selection of a model is contingent on specific requirements. Given sufficient computational resources, opting for YOLOv7 or YOLOv6l could be advantageous for achieving high performance. Conversely, when faced with limitations in computation power, the consideration of lighter models like YOLOv6s or YOLOv5s may be warranted.

The results can be displayed in the following Table 3 when considering the time of execution.

Table 3. Model Execution Time

Algorithm	Time		
	Average pre-process	Average inference	Average NMS
YOLOv7	0.25 ms	112.9 ms	1.0 ms
YOLOv6l	0.18 ms	45.68 ms	1.97 ms
YOLOv6m	0.18 ms	27.43 ms	1.51 ms
YOLOv6s	0.18 ms	12.72 ms	2.52 ms

YOLOv5l	0.4 ms	40.6 ms	1.6 ms
YOLOv5m	0.4 ms	22.0 ms	1.6 ms
YOLOv5s	0.4 ms	9.1 ms	1.2 ms

YOLOv7 performs well in terms of accuracy (mAP) and requires a reasonable inference time (112.9 ms). The low pre-processing time (0.25 ms) indicates that the model processes the input quickly before going through the inference process.

YOLOv6l performs reasonably well in terms of accuracy, although it requires more computational power than YOLOv7. The slower inference time (45.68 ms) indicates that this model is slower in detecting objects. YOLOv6m is a more efficient model in terms of computation power than YOLOv6l but with a slight sacrifice in accuracy. The faster inference time (27.43 ms) could be viable when working with restricted computational resources. YOLOv6s is the lightest model regarding computation power, but accuracy and inference time are lower. The high NMS time (2.52 ms) may be the reason for the longer inference time.

YOLOv5l has good performance in terms of accuracy and reasonable inference time. However, the slightly higher pre-processing time (0.4 ms) and slower inference time (40.6 ms) should be considered. YOLOv5m is a lighter model with lower performance compared to the larger model. The faster inference time (22.0 ms) can be a good choice for real-time applications. YOLOv5s is a very lightweight model with a fast inference time (9.1 ms), though slightly lower accuracy.

4.3 Model Evaluation

Table 4**.** shows the analysis results performed on each model using testing data.

Table 4. Model Evaluation Metrics

Algorithm	Metrics Evaluation		
	Accuracy	AP	AR
YOLOv7	0,51	0,500	0,513
YOLOv6l	0,37	0,379	0,371
YOLOv6m	0,33	0,399	0,326
YOLOv6s	0,37	0,401	0,365
YOLOv5l	0,39	0,401	0,391
YOLOv5m	0,39	0,406	0,385
YOLOv5s	0,43	0,492	0,431

Based on Table 4. evaluation metrics, YOLOv7 has high accuracy and good performance in terms of average precision (AP) and average recall (AR). However, despite its fast inference time, it requires higher computational power (GFLOPs: 103.2) than other models.

YOLOv6l has lower accuracy than YOLOv7 and lower AP and AR performance. However, its inference time is faster than that of YOLOv7. YOLOv6m has lower accuracy but slightly higher AP performance than YOLOv6l. The inference time is faster than YOLOv6l but still requires significant computation power. YOLOv6s has a similar accuracy level to YOLOv6l but with a faster inference time. It is a good choice when seeking a balance between accuracy and speed of inference.

YOLOv5l has a relatively high accuracy rate and good AP and AR performance. Its moderate inference time can also be a good choice if it has sufficient resources. YOLOv5m has fairly good accuracy and faster inference time compared to other models. It can be a good choice for real-time applications. While the YOLOv5s have good accuracy and very fast inference time. This model can be a strong choice for real-time applications that require high-speed object detection.

4.4 Best Model Evaluation

Based on the experiments and analysis that have been carried out, many factors are decisive in determining the best model. After further observation, the two best models are YOLOv5s and YOLOv7.

The selection of YOLOv5s is based on a good level of accuracy (Accuracy: 0.43) with a competitive mAP (mAP@50: 0.549). Very fast inference time (Average inference: 9.1 ms), making it suitable for real-time applications. It also has a very low GFLOPs rate (45,15), which means it requires low computation power.

Meanwhile, YOLOv7 has a higher accuracy rate (Accuracy: 0.51) with a good mAP (mAP@50: 0.559). Reasonable inference time (Average inference: 112.9 ms), which may still be usable for real-time applications depending on the requirement. Although it requires higher computation power (GFLOPs: 103.2), it is a good choice if accuracy is a top priority.

The model evaluation is more detailed, using a matrix of confusion, which is more closely related to the best model. The following Table 5 shows the comparison results of the test.

Table 5. Confusion Matrix YOLOv5s

Actual	Predicted					
		CS	EP	GC	LF	N	BG
	CS	0,69	0	0	0,11	0,14	0,06
	EP	0,06	0,25	0,03	0,56	0	0,11
	GC	0,03	0	0,11	0,57	0,17	0,13
	LF	0,04	0	0	0,78	0,11	0,05
	N	0,08	0	0,01	0,14	0,75	0,03
	BG	0,25	0,03	0,02	0,39	0,31	0

From Table 5. confusion matrix, there are five active classes, namely Cheat Sheet (CS), Exchange Paper (EP), Giving Code (GC), Looking Friend (LF) and Normal (N). At the same time, BG is a Background (BG) class that describes the part of the image that does not contain objects of any class identified by the model. It is a specialized class used to indicate areas without significant or important objects in the image.

Table 6. Confusion Matrix YOLOv7

Actual	Predicted					
		CS	EP	GC	LF	N	BG
	CS	0,72	0	0	0,09	0,16	0,02
	EP	0,04	0,52	0,04	0,17	0,13	0,09
	GC	0,06	0	0,28	0,34	0,29	0,03
	LF	0,05	0,02	0,04	0,74	0,13	0,03
	N	0,07	0	0,01	0,1	0,81	0,01
	BG	0,21	0,04	0,08	0,36	0,31	0

Based on the analysis of the confusion matrix in Table 5 and Table 6, it can be concluded that the YOLOv7 model performs better than the YOLOv5s model in detecting the Exchange Paper and Giving Code classes in the imbalance dataset case. In the Exchange Paper class, YOLOv7 has a True Positive (TP) value of 0.52, while YOLOv5s only has a TP of 0.19. Similarly, in the Giving Code class, YOLOv7 has a TP of 0.28, while YOLOv5s only has a TP of 0.06.

Nonetheless, it is important to note that the performance of both models is far from ideal. This may be due to several factors. Firstly, the dataset's imbalance may affect the model's performance, as the model may tend to focus on the majority class and ignore the minority class. Secondly, the parameters and hyperparameters of the model may play an important role in performance. In further research, parameter tuning, and more careful dataset processing may be required to improve model performance. In addition, the performance of cheating detection can also be affected by environmental factors, such as the quality of CCTV images and the intelligence level of the model used.

4.5 Adding Dataset Experiment

After obtaining YOLOv7 as the best-performing model in the previous experiments, the researcher endeavored to further train the YOLOv7 model by adding 10% data for two labels that exhibited data imbalance, namely "Giving Code" (592 data) and "Exchange Paper" (525 data). Consequently, a new dataset was acquired as follows:

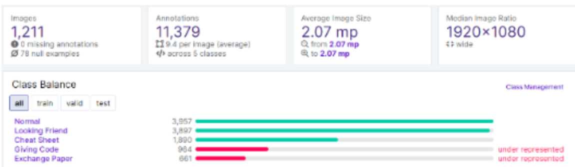

Fig. 7. Add Giving Code and Exchange Paper Dataset

The training process followed the same methodology as the previous training, with no additional changes to the training or testing scenarios. Upon completion of the training, an accuracy value of 0.60 and a mAP of 0.66 were obtained. This indicates an improvement in accuracy of approximately 9% and an increase in mAP of around 10% compared to the previous experiment.

From this experiment, it is evident that augmenting the dataset for classes with data imbalance can enhance the accuracy performance of the model, resulting in improved outcomes. The increased dataset size contributes to a more comprehensive representation of the specific classes, allowing the model to learn better and generalize patterns associated with Giving Code and Exchange Paper.

5 Conclusion

The best models to use are YOLOv5s and YOLOv7, according to the tests carried out in this study. The YOLOv5s model is a very lightweight model with a very fast inference time (9.1 ms) and does not require large computations in its implementation. On the other hand, this model has a significant shortcoming when detecting classes with dataset deficiencies, namely Giving Code and Exchange Paper. However, this deficiency can be recognized better in YOLOv7 with better accuracy of 51%, an increase of about 8% higher than the YOLOv5s model, so the YOLOv7 model is the best option in the first scenario.

In the second scenario, models were constructed by adding the dataset by 10%, focusing on addressing the imbalance in two classes, Giving Code and Exchange Paper. Utilizing YOLOv7 yielded an accuracy of 60%, indicating a notable improvement of 9% compared to the previous model. This observation underscores the significant impact of data imbalance on model performance.

Future development can be done by adding datasets to classes with deficiencies to obtain a balanced dataset. Additionally, exploring alternative research methods, such as experimenting with different model architectures and incorporating advanced algorithms, presents opportunities for novel insights into improving detection accuracy and robustness. Fine-tuning model hyperparameters and systematically optimizing factors like learning rates and anchor box sizes are crucial for enhancing adaptability. Furthermore, evaluating models in more realistic scenarios, including diverse environmental conditions and complex backgrounds, ensures their applicability beyond controlled experiments. This multifaceted approach aims to propel YOLOv7 and similar models toward heightened accuracy, adaptability, and practical utility in real-world contexts.

References

[1] H. Cai and I. King, “Education Technology for Online Learning in Times of Crisis,” in *Proceedings of 2020 IEEE International Conference on Teaching, Assessment, and Learning for Engineering, TALE 2020*, Institute of Electrical and Electronics Engineers Inc., Dec. 2020, pp. 758–763. doi: 10.1109/TALE48869.2020.9368387.

[2] L. Al-Labadi and S. Sant, “Enhance Learning Experience Using Technology In Class,” *J Technol Sci Educ*, vol. 11, no. 1, pp. 44–52, Sep. 2021, doi: 10.3926/jotse.1050.

[3] M. Alahmari, “Exploring the Influential Factors Affecting Staff Willingness to Adopt Augmented Reality,” *International Journal of Information and Education Technology*, vol. 13, no. 7, pp. 1078–1084, Jul. 2023, doi: 10.18178/ijiet.2023.13.7.1907.

[4] S. R. Sletten, “Rethinking Assessment: Replacing Traditional Exams with Paper Reviews,” *J Microbiol Biol Educ*, vol. 22, no. 2, Sep. 2021, doi: 10.1128/jmbe.00109-21.

[5] É. Cambron-Goulet, J. P. Dumas, É. Bergeron, L. Bergeron, and C. St-Onge, “Guidelines for Creating Written Clinical Reasoning Exams: Insight from a Delphi Study,” *Health Professions Education*, vol. 5, no. 3, pp. 237–247, Sep. 2019, doi: 10.1016/j.hpe.2018.09.001.

[6] R. Ellis, J. Cleland, D. SG. Scrimgeour, A. J. Lee, J. Hines, and P. A. Brennan, “Establishing the predictive validity of the intercollegiate membership of the Royal Colleges of surgeons written examination: MRCS Part A,” *The Surgeon*, Aug. 2023, doi: 10.1016/j.surge.2023.07.004.

[7] M. J. Hoque, Md. Razu, Md. Jashim, and M. Mostafa, “Automation of Traditional Exam Invigilation using CCTV and Bio-Metric,” *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 6, pp. 392–399, 2020, doi: 10.14569/IJACSA.2020.0110651.

[8] R. E. Ferdig, E. Baumgartner, R. Hartshorne, R. Kaplan-Rakowski, and C. Mouza, “Teaching, Technology, and Teacher Education During the COVID-19 Pandemic: Stories From The Field,” *AACE-Association for the Advancement of Computing in Education*, pp. 125–128, Jun. 2020.

[9] R. Comas-Forgas, T. Lancaster, A. Calvo-Sastre, and J. Sureda-Negre, “Exam cheating and academic integrity breaches during the COVID-19 pandemic: An analysis of internet search activity in Spain,” *Heliyon*, vol. 7, no. 10, Oct. 2021, doi: 10.1016/j.heliyon.2021.e08233.

[10] https://academicintegrity.org/, “THE FUNDAMENTAL VALUES OF ACADEMIC INTEGRITY,” International Center for Academic Integrity. Accessed: Sep. 09, 2023. [Online]. Available: https://academicintegrity.org/images/pdfs/20019_ICAI-Fundamental-Values_R12.pdf

[11] M. C. Ossai, N. Ethe, D. E. Edougha, and O. D. Okeh, “Academic integrity during examinations, age and gender as predictors of academic performance among high school students,” *Int J Educ Dev*, vol. 100, Jul. 2023, doi: 10.1016/j.ijedudev.2023.102811.

[12] J. S. Hutton, J. Dudley, T. Horowitz-Kraus, T. DeWitt, and S. K. Holland, “Associations between home literacy environment, brain white matter integrity and cognitive abilities in preschool-age children,” *Acta Paediatrica, International Journal of Paediatrics*, vol. 109, no. 7, pp. 1376–1386, Jul. 2020, doi: 10.1111/apa.15124.

[13] S. Janke, S. C. Rudert, Ä. Petersen, T. M. Fritz, and M. Daumiller, “Cheating in the wake of COVID-19: How dangerous is ad-hoc online testing for academic integrity?,” *Computers and Education Open*, vol. 2, p. 100055, Dec. 2021, doi: 10.1016/j.caeo.2021.100055.

[14] D. Septiana, “HUBUNGAN EFIKASI DIRI DAN STRES AKADEMIK DENGAN PERILAKU MENYONTEK SISWA,” UNIVERSITAS ISLAM NEGERI RADEN INTAN LAMPUNG, Lampung, 2022. Accessed: Nov. 18, 2023. [Online]. Available: http://repository.radenintan.ac.id/17318/

[15] A. Fadillah, “Hubungan Antara Efikasi Diri Dengan Perilaku Menyontek Pada Mahasiswa,” *Psikoborneo: Jurnal Ilmiah Psikologi*, vol. 7, no. 4, pp. 657–664, 2019.

[16] Zulkifli N, “Efektivitas Pelaksanaan Ujian Semester Menggunakan Computer Based Test,” *Edumaspul: Jurnal Pendidikan*, vol. 6, no. 1, pp. 186–191, Mar. 2022, doi: 10.33487/edumaspul.v6i1.3057.

[17] B. Chen, S. Azad, M. Fowler, M. West, and C. Zilles, “Learning to Cheat: Quantifying Changes in Score Advantage of Unproctored Assessments Over Time,” in *L@S 2020 - Proceedings of the 7th ACM Conference on Learning @ Scale*, Association for Computing Machinery, Aug. 2020, pp. 197–206. doi: 10.1145/3386527.3405925.

[18] C. M. Ifeagwazi, J. C. Chukwuorji, S. O. Egbodo, and M. B. Nwoke, “Peer pressure, fear of failure and examination cheating behavior in the university: Does gender make the difference?,” *Cognition, Brain, Behavior. An Interdisciplinary Journal* , vol. 23, no. 1, pp. 43–62, 2019, doi: 10.24193/cbb.2019.23.03.

[19] A. Siddhpura and M. Siddhpura, “Plagiarism, contract cheating and other academic misconducts in online engineering education: Analysis, detection and prevention strategies,” in *Proceedings of 2020 IEEE International Conference on Teaching, Assessment, and Learning for Engineering, TALE 2020*, Institute of Electrical and Electronics Engineers Inc., Dec. 2020, pp. 112–119. doi: 10.1109/TALE48869.2020.9368311.

[20] M. Mahajan, R. Rajeshwaribajre, L. Sharma, and S. L. Varma, “Computer Vision Based Campus Surveillance,” in *Proceedings - IEEE 2020 2nd International Conference on Advances in Computing, Communication Control and Networking, ICACCCN 2020*, Institute of Electrical and Electronics Engineers Inc., Dec. 2020, pp. 1013–1018. doi: 10.1109/ICACCCN51052.2020.9362776.

[21] J. Nishchal, S. Reddy, and P. N. Navya, “Automated Cheating Detection in Exams using Posture and Emotion Analysis,” in *2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, IEEE, Jul. 2020, pp. 1–6. doi: 10.1109/CONECCT50063.2020.9198691.

[22] C. Wu, Q. Luo, and D. Shou, “Image Recognition Technology and Development of YOLO Algorithm,” in *Advances in Transdisciplinary Engineering*, IOS Press BV, Dec. 2022, pp. 32–38. doi: 10.3233/ATDE221146.

[23] T. Diwan, G. Anirudh, and J. V. Tembhurne, "Object detection using YOLO: challenges, architectural successors, datasets and applications," *Multimed Tools Appl*, vol. 82, no. 6, pp. 9243–9275, Mar. 2023, doi: 10.1007/s11042-022-13644-y.

[24] V. Senthilkumar, G. Kanagaraj, T. Primya, J. Joycema, M. B. Joan, and J. A. Vicram, "Application of AI And Computer Vision To Face Mask And Social Distance Detection in CCTV Video Streams," in *2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA)*, IEEE, Oct. 2021, pp. 1–5. doi: 10.1109/ICAECA52838.2021.9675746.

[25] E. P. Ijjina, A. S. Joshi, and G. Kanahasabai, "Detection of Customer Interested Garments in Surveillance Video using Computer Vision," in *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Kharagpur, India: IEEE, Jul. 2020, pp. 1–5. doi: 10.1109/ICCCNT49239.2020.9225571.

[26] A. Srinivasan, A. Srikanth, H. Indrajit, and V. Narasimhan, "A Novel Approach for Road Accident Detection using DETR Algorithm," in *2020 International Conference on Intelligent Data Science Technologies and Applications, IDSTA 2020*, Institute of Electrical and Electronics Engineers Inc., Oct. 2020, pp. 75–80. doi: 10.1109/IDSTA50958.2020.9263703.

[27] N. Chhadikar, P. Bhamare, K. Patil, and S. Kumari, "Image processing based Tracking and Counting Vehicles," in *2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)*, Coimbatore, India: IEEE, Jun. 2019, pp. 335–339. doi: 10.1109/ICECA.2019.8822070.

[28] X. Cong, S. Li, F. Chen, C. Liu, and Y. Meng, "A Review of YOLO Object Detection Algorithms based on Deep Learning," *Frontiers in Computing and Intelligent Systems*, vol. 4, no. 2, pp. 17–20, Jun. 2023, doi: 10.54097/fcis.v4i2.9730.

[29] Institute of Electrical and Electronics Engineers. Bangalore Section and Institute of Electrical and Electronics Engineers, *Automated Cheating Detection in Exams using Posture and Emotion Analysis*.

[30] M. M. Masud, K. Hayawi, S. S. Mathew, T. Michael, and M. El Barachi, "Smart Online Exam Proctoring Assist for Cheating Detection," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Science and Business Media Deutschland GmbH, Jan. 2022, pp. 118–132. doi: 10.1007/978-3-030-95405-5_9.

[31] Y. Fang, J. Ye, and H. Wang, "Realization of Intelligent Invigilation System Based on Adaptive Threshold," in *2020 5th International Conference on Computer and Communication Systems (ICCCS)*, IEEE, May 2020, pp. 201–205. doi: 10.1109/ICCCS49078.2020.9118581.

[32] C. Liu, H. Zhou, H.-C. Xu, B.-Y. Hou, and L. Yao, "Abnormal Behavior Recognition in an Examination Based on Pose Spatio-Temporal Features," in *2020 IEEE 5th International Conference on Cloud Computing and Big Data*

Analytics (ICCCBDA), IEEE, Apr. 2020, pp. 380–386. doi: 10.1109/ICCCBDA49378.2020.9095710.

[33] M. Adil, R. Simon, and S. K. Khatri, "Automated Invigilation System for Detection of Suspicious Activities during Examination," in *2019 Amity International Conference on Artificial Intelligence (AICAI)*, IEEE, Feb. 2019, pp. 361–366. doi: 10.1109/AICAI.2019.8701263.

[34] Kulkarni Rutuja Sanjay, "Real Time Automated Invigilator in Classroom Monitoring using Computer Vision," *2nd International Conference on Advances in Science & Technology (ICAST-2019)*, p. 396, Apr. 2019, doi: https://dx.doi.org/10.2139/ssrn.3367715.

[35] S. Kaddoura and A. Gumaei, "Towards effective and efficient online exam systems using deep learning-based cheating detection approach," *Intelligent Systems with Applications*, vol. 16, Nov. 2022, doi: 10.1016/j.iswa.2022.200153.

[36] F. Hussein, A. Al-Ahmad, S. El-Salhi, E. Alshdaifat, and M. Al-Hami, "Advances in Contextual Action Recognition: Automatic Cheating Detection Using Machine Learning Techniques," *Data (Basel)*, vol. 7, no. 9, Sep. 2022, doi: 10.3390/data7090122.

[37] M. Malhotra and I. Chhabra, "Automatic Invigilation Using Computer Vision," in *Proceedings of the 3rd International Conference on Integrated Intelligent Computing Communication & Security (ICIIC 2021)*, Sep. 2021, pp. 130–136. doi: https://doi.org/10.2991/ahis.k.210913.017.

[38] A. Zaffar, M. Jawad, M. Shabbir, P. O. Factory, and W. Cantt, "A Novel CNN-RNN Model for E-Cheating Detection Based on Video Surveillance," *UW JOURNAL OF CMPUTER SCIENCE*, vol. 5, no. 1, Jun. 2023, Accessed: Sep. 11, 2023. [Online]. Available: https://uwjcs.org.pk/index.php/ojs/article/view/64

[39] J. E. Ball, D. T. Anderson, and C. S. C. Sr., "Comprehensive survey of deep learning in remote sensing: theories, tools, and challenges for the community," *J Appl Remote Sens*, vol. 11, no. 4, p. 42609, 2017, doi: 10.1117/1.JRS.11.042609.

[40] B. Setiyono, D. A. Amini, and D. R. Sulistyaningrum, "Number plate recognition on vehicle using YOLO - Darknet," in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Mar. 2021. doi: 10.1088/1742-6596/1821/1/012049.

[41] S. K. Shandilya, A. Srivastav, K. Yemets, A. Datta, and A. K. Nagar, "YOLO-based segmented dataset for drone vs. bird detection for deep and machine learning algorithms," *Data Brief*, vol. 50, p. 109355, Oct. 2023, doi: 10.1016/j.dib.2023.109355.

[42] E. D. and G. G. and L. T.-Y. and S. J. and L. Q. V Zoph Barret and Cubuk, "Learning Data Augmentation Strategies for Object Detection," in *Computer Vision – ECCV 2020*, H. and B. T. and F. J.-M. Vedaldi Andrea and Bischof, Ed., Cham: Springer International Publishing, 2020, pp. 566–583.

[43] P. Sharma, P. Hans, and S. C. Gupta, "Classification Of Plant Leaf Diseases Using Machine Learning And Image Preprocessing Techniques," in *2020*

10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), IEEE, Jan. 2020, pp. 480–484. doi: 10.1109/Confluence47617.2020.9057889.

[44] Z. Q. Zhao, P. Zheng, S. T. Xu, and X. Wu, "Object Detection with Deep Learning: A Review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11. Institute of Electrical and Electronics Engineers Inc., pp. 3212–3232, Nov. 01, 2019. doi: 10.1109/TNNLS.2018.2876865.

[45] Meituan, "YOLOv6 By Meituan Vision AI Department," Meituan Vision AI Department. Accessed: Sep. 18, 2023. [Online]. Available: https://github.com/meituan/YOLOv6

[46] O. E. Olorunshola, M. E. Irhebhude, and A. E. Evwiekpaefe, "A Comparative Study of YOLOv5 and YOLOv7 Object Detection Algorithms," *Journal of Computing and Social Informatics*, vol. 2, no. 1, pp. 1–12, Feb. 2023, doi: 10.33736/jcsi.5070.2023.

[47] L. Barba-Guaman, J. E. Naranjo, A. Ortiz, and J. G. P. Gonzalez, "Object Detection in Rural Roads Through SSD and YOLO Framework," in *Trends and Applications in Information Systems and Technologies*, H. and D. G. and M. F. and R. C. A. M. Rocha Álvaro and Adeli, Ed., Cham: Springer International Publishing, 2021, pp. 176–185. doi: 10.1007/978-3-030-72657-7_17.

[48] Z. Soleimanitaleb, M. A. Keyvanrad, and A. Jafari, "Object Tracking Methods:A Review," in *2019 9th International Conference on Computer and Knowledge Engineering (ICCKE)*, IEEE, Oct. 2019, pp. 282–288. doi: 10.1109/ICCKE48569.2019.8964761.

[49] M. Hasnain, M. F. Pasha, I. Ghani, M. Imran, M. Y. Alzahrani, and R. Budiarto, "Evaluating Trust Prediction and Confusion Matrix Measures for Web Services Ranking," *IEEE Access*, vol. 8, pp. 90847–90861, May 2020, doi: 10.1109/ACCESS.2020.2994222.

[50] R. Ilahi, F. Bimantoro, R. Dwiyansaputra, and R. Farinda, "Classification of Pringgasela Typical Songket Using Multi Texton Co-occurrence Descriptor and K-Nearest Neighbor," in *Proceedings of the First Mandalika International Multi-Conference on Science and Engineering 2022, MIMSE 2022 (Informatics and Computer Science)*, Dordrecht: Atlantis Press International BV, 2022, pp. 352–366. doi: 10.2991/978-94-6463-084-8_30.

[51] G. Tripathi and R. Kumar, "Early Prediction of Diabetes Mellitus Using Machine Learning," in *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, IEEE, Jun. 2020, pp. 1009–1014. doi: 10.1109/ICRITO48877.2020.9197832.

[52] L. Farokhah, "IMPLEMENTASI K-NEAREST NEIGHBOR UNTUK KLASIFIKASI BUNGA DENGAN EKSTRAKSI FITUR WARNA RGB," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, vol. 7, no. 6, pp. 1129–1135, 2019, doi: 10.25126/jtiik.202072608.

[53] J. Lee and K. il Hwang, "YOLO with adaptive frame control for real-time object detection applications," *Multimed Tools Appl*, vol. 81, no. 25, pp. 36375–36396, Oct. 2022, doi: 10.1007/s11042-021-11480-0.

[54] M. Durve, F. Bonaccorso, A. Montessori, M. Lauricella, A. Tiribocchi, and S. Succi, “Tracking droplets in soft granular flows with deep learning techniques,” *The European Physical Journal Plus*, vol. 136, no. 8, p. 864, Aug. 2021, doi: 10.1140/epjp/s13360-021-01849-3.

[55] D. Gros *et al.*, “Anaphora Resolution,” *International Journal of Information Retrieval Research*, vol. 8, no. 3, pp. 33–45, Jul. 2018, doi: 10.4018/IJIRR.2018070103.

[56] M. Zhu, “Recall, Precision and Average Precision,” Canada, Aug. 2004. Accessed: Sep. 14, 2023. [Online]. Available: https://www.researchgate.net/publication/228874142_Recall_precision_and_average_precision

[57] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, “A Review of Yolo Algorithm Developments,” in *Procedia Computer Science*, Elsevier B.V., 2021, pp. 1066–1073. doi: 10.1016/j.procs.2022.01.135.

[58] S. Xu *et al.*, “PP-YOLOE: An evolved version of YOLO,” Mar. 2022, [Online]. Available: http://arxiv.org/abs/2203.16250

[59] Afis Julianto, Andi Sunyoto, and Ferry Wahyu Wibowo, “OPTIMASI HYPERPARAMETER CONVOLUTIONAL NEURAL NETWORK UNTUK KLASIFIKASI PENYAKIT TANAMAN PADI,” *TEKNIMEDIA: Teknologi Informasi dan Multimedia*, vol. 3, no. 2, pp. 98–105, Dec. 2022, doi: 10.46764/teknimedia.v3i2.77.

[60] 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, Nov. 2018, pp. 1–4. doi: 10.1109/PUNECON.2018.8745376.

[61] Q. H. Nguyen *et al.*, “Influence of Data Splitting on Performance of Machine Learning Models in Prediction of Shear Strength of Soil,” *Math Probl Eng*, vol. 2021, pp. 1–15, Feb. 2021, doi: 10.1155/2021/4832864.

[62] G. Varma, A. Subramanian, A. Namboodiri, M. Chandraker, and C. V. Jawahar, “IDD: A Dataset for Exploring Problems of Autonomous Navigation in Unconstrained Environments,” in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, Jan. 2019, pp. 1743–1751. doi: 10.1109/WACV.2019.00190.