An Efficient Projection Screen Detection from Presentation Videos using CustomYOLOv7 Object Detection Method

E. Purushotham (purushothame882@gmail.com)
JNTUK
Kasarapu Ramani
Sree vidyanikethan engineering college
C. Shobha Bindu
JNTUA College of Engineering

Research Article

Keywords: Object Detection, Projection Screen Detection, YOLOv7, Presentation Video, Retinanet

Posted Date: January 16th, 2024

DOI: https://doi.org/10.21203/rs.3.rs-3852536/v1

Additional Declarations: No competing interests reported.

Abstract

Lecture videos are analyzed for the development of various applications that involves indexing, summarization, content extraction, search, and navigation. Lecture videos captured in classrooms and conference rooms has digital slides projected on to the screen on stage. Projection screen detection is a crucial task for the extraction of slide region in such presentation videos. In this paper, we present an interesting approach for detecting the location of slide region in video frames using the You Only Look Once (YOLO) object detection model. First we train the custom YOLOv7 model on a labelled dataset of frames from presentation videos showing projected slides and then the trained model is applied on new images that are not used in training to predict the location of projector screens. We collected and annotated over 2000 frames from various presentation videos and then various augmented techniques are applied to prepare a dataset of 5000 images. We evaluated this method on our custom dataset and the results are compared with other popular object detection methods. Our experiments demonstrated that our custom YOLOv7 model outperforms basic YOLOv7 and Retinanet with regards to accuracy and computational effectiveness. Our results suggest that custom YOLOv7 provides a promising solution for projector screen detection and has the potential to be applied in various practical applications

1. Introduction

Every day, Countless educational videos are captured inside classrooms and conference halls across the globe. Lecture videos that capture the classroom interaction offer an authentic learning experience to students who are unable to attend the classes physically. However, most of these videos are unstructured and lengthy, which makes it challenging to efficiently index, browse, navigate, and search for content.

Lecture videos are analyzed for the development of various applications that utilize the content of lecture videos, such as indexing [1–5], summarization [6–11], content extraction [12–17], search [18–21], and navigation [22–25]. Lecture videos captured in classrooms and conference rooms has digital slides projected on to the screen on stage, a common setup in modern classrooms and conference rooms. These videos have projected slide, presenter explaining the slides and audience in single frame, the primary element of a learning environment is the slide screen, which contains most of the important information. The text within the slides is a critical component for indexing and retrieving video content, which necessitates the need for an efficient slide detector to extract the slide contents. In lecture videos, the slide region is a crucial object that carries essential information. Therefore, it is necessary to extract and segment the slide region from other objects in the scene

In this paper, we focus on lecture videos that involve digital slides projected onto a screen on stage, a typical arrangement found in contemporary classrooms and conference rooms as shown in Fig. 1. It is assumed that camera with zoom, tilt, and pan capabilities is used to record video. To identify the slide region, we used YOLO object detection technique to automatically detect the location of the projector screen in the lecture video in order to identify the slide region.

This paper presents a study on projection screen detection in presentation videos using YOLOv7 [26], an object detection model. To evaluate the performance of YOLOv7, a dataset of frames is collected from various presentation videos with complex illumination conditions and camera movements. Our experiments demonstrate that YOLOv7 outperforms RetinaNet in terms of speed and accuracy, indicating that it is a viable option for projection screen recognition in presentation videos. Our findings can be beneficial for professionals and researchers engaged in video analysis and object detection.

In this study, we use the terms 'projected screen' and 'projected slide region' interchangeably to refer to the area of the screen on which the presentation slides are displayed.

The following are our contributions made in this research.

1) we collected and annotated a dataset of 2093 frames from various presentation videos with complex illumination conditions and camera movements. From the collected dataset another new dataset of 5000 images is created by applying various augmented techniques.

2) A custom YOLOv7 model is made by changing the default image grid size from 32 to 64 and disabling the mosaic augmentation feature of YOLOv7.

3)Trained the custom YOLOv7, basicYOLOv7 and Retinanet [27] models using transfer learning techniques.

4) An evaluation showing customYOLOv7 method outperforms basicYOLOv7 and Retinanet models on the new dataset.

2. Related work

Kai li et al. [28] developed a system that can automatically extract the semantic structure of a video presentation in an academic setting. In the video, it can identify and monitor both the presenter and the projection screen. By analyzing the visual elements present in the tracked area of the screen, the system detects slide progressions and extracts a list of high-quality images that effectively capture the key topics of the presentation. The system uses sparse feature points for projection screen localization, instead of using raw pixels

Rajgure et al., [29] suggested a method for localizing slides within a presentation video. To distinguish the slide from the background, the method applies a quick segmentation technique in the video frame's marginal space. The slide can be localized without having any prior knowledge about the slide's a and color.

Soundes et al. [30] have proposed an approach that partition the video frame into equal sized windows for calculating pseudo Zernike moments, which act as local features. The k-means technique is applied on the detected features to cluster them. A shape recognizing step is performed to identify the location of the slide and isolated from the original frame.

Zhao et al. [31] proposed a ROI (Region of Interest) ranking approach for automatically recognizing slides inside educational videos. First the video is partitioned into shots and classify them into two categories using image processing analysis and machine learning algorithms. Each frame is binarized using range of thresholds and edge features are extracted from each shot to find slide regions within a frame.

Thomas et al. [32] proposed an automated method for predicting the instructor's style of presentation and level of student engagement in educational videos. This technique examined a number of visual and acoustic elements that were taken out of the video frames, including motion, face expressions, and speech patterns. A machine learning model is trained to predict the nature of the presentation, as well as the degree of student participation. The slide identification model generates a bounding box around each image's slide area using all of the video frames as input in order to identify the slide region in a lecture video. The identified slide area is then used in a later study. The RetinaNet model was employed in the slide area detection process.

The problem addressed in this paper is the detection of projected slide in presentation videos. The task involves identifying the location of the slide region in each frame of the video. Variations in lighting, camera angles and occlusions makes the problem more challenging. The objective is to identify an accurate and efficient object detection model that can automatically detect projection screens in real-world presentation videos.

Inspired from the paper from Thomas et al. [32], when Retinanet model used for slide region object detection on the frames of classroom/conference room videos, the model was not able to detect the slide region completely. The Retinanet model was able to recognize slide region in the frame, but bounding box is not covering the complete slide region for many of the images. In this paper we test a custom YOLOv7 object detection model, for detecting the slide regions in lecture videos captured in classroom and conference rooms.

3. Methodology

Based on the problem definition, the methodology involves training a YOLOv7 object detection model on a new dataset consisting of frames from presentation videos to detect the projection screen. The work flow is as shown in the Fig. 2

3.1. Dataset:

3.1.1. Data Collection:

As there is no bench mark dataset of frames from lecture videos captured in classrooms and conference rooms, a total of 225 videos are collected from different sources like ClassX dataset[33], videolectures.net, NPTEL, Youtube channels of MITopencourseware, yale university and Villanova university, and various conference, seminar and viva-voce videos from YouTube. Frames are extracted from these videos and 2093 unique frames with different environments and different lighting conditions are selected as dataset.

3.1.2. Data Preprocessing:

Next, preprocessing was done on the videos to extract frames and 2093 distinct frames with different environments and different lighting conditions are selected. All the selected frames are resized to 640X640, corresponding bounding boxes around the projection screens are manually annotated using the roboflow[34], the Fig. 2. Shows the sample images labelled using roboflow.

3.1.3. Data Augmentation:

Data augmentation confers several benefits [35], including the enhancement of a model's ability to generalize, introduction of variability into the data to mitigate overfitting, reduction of the cost of collecting and annotating additional data, and improvement in the accuracy of the predictions made by deep learning models. So Some data augmentation techniques, including cropping with 0% minimum zoom and 15% maximum zoom, rotation within − 5° to + 5°, brightness and exposure adjustment between − 25% and + 25(both for image and bounding box), were utilized as a result dataset is populated with 5000 images in total with 2093 original images and the remaining augmented images.

Two datasets are considered for experimentation, one with 2093 original images and the second with 5000 images. First dataset is split into 70% for training, 20% for validation and 10% for testing, the second dataset has the actual validation and testing set from first dataset and the original training set plus its augmented images for training.

3.2. Modelling:

In this paper the modified YOLOv7 model with custom hyperparameters is used. The YOLOv7[25] model belongs to the You Only Look Once (YOLO) group of models, which are single-stage object detectors. In this type of model, the image frames are first processed by a backbone to obtain features. These features are then combined and processed by the neck before being passed to the head of the network. Utilizing this architecture, the YOLOv7 model predicts the object class and location within the image, and subsequently draws bounding boxes around the identified objects.

Several structural changes were made to YOLOv7 by the developers, including the extended efficient layer aggregation network (EELAN), compound scaling, and the addition of planned and reparametrized convolution as one of a variety of supplementary methods. They also added fineness for lead loss and coarseness for auxiliary loss as extra improvements.

The foundation of YOLOv7 is its EELAN architecture, which, in order to improve its learning capacities without sacrificing the integrity of the original gradient path, applies the novel "expand, shuffle, and merge cardinality" technique.

Model scaling's main goal is alter significant model attributes to generate models that meet various application requirements. The primary objective of model scaling is to. Modifications to the model's scale can improve dimensions, depth, and resolution. Scaling parameters are interdependent, therefore in

traditional methods that use concatenation-based architectures (like PlainNet or ResNet), it is necessary to take them into account at the same time. For example, changing the depth of the model will modify the input-to-output channel ratio of a transition layer, this could have an impact on the hardware requirements of the model. As a result, for concatenation-based models, YOLOv7 uses a compound scaling technique that allows for the maintenance of the optimal design and the preservation of the model's original attributes[25].

Reparameterization is a post-training augmentation method that improves inference results while prolonging the training process. To maximize model performance, two main reparameterization techniques are frequently used: model-level ensembles and module-level ensembles. Furthermore, a convolutional block variant known as RepConv is adopted without identity connection (RepConvN) within the reparametrized architecture of YOLOv7 [25].

In the YOLO architecture, the model output is presented by the 'head' module. In YOLOv7, the auxiliary head facilitates training in the intermediate layers, while the lead head produces the final output. A two-label assigner approach facilitates an optimal training process where soft labels are produced by the lead head guided label assigner using the ground truth and the lead head's predictions. These soft labels are used for training by both the lead and auxiliary heads, with the lead head's labels providing a more thorough depiction of the data distribution and correlation. Furthermore, the coarse-to-fine lead head directed label assigner generates both coarse and fine labels, where the former loosens restrictions to maximize the auxiliary head's recall in object detection. By dynamically adjusting the importance of the fine and coarse labels during the learning process, the method ensures an optimizable upper bound for the fine label, maintaining a balance between the two labels to enhance the overall learning process [25].

As per our dataset the projection screen objects are larger in size, we increased the stride size to 64 from 32 as a result the output feature map size will decrease, and the feature map cell count will be reduced. As result it could potentially improve the detection of larger objects, decreasing the number of cells can also result in a faster model inference time, as there are fewer cells to process.

We used the YOLOv7 obtained from GitHub repository[36] and made the changes to stride size.

3.3. Model training:

The pre-trained weights of the YOLOv7 model on the COCO dataset were used for transfer learning on the new dataset with the following modification to hyperparameters. Default mosaic and mixup options of basic YOLOv7 are disabled, batch size is set to 16 and number of training epochs is set to 25. The remaining hyperparameters kept to their default values.

Algorithm for training one epoch of YOLOv7 model

Step1. Shuffle the training dataset.

Step2. For each batch in the training dataset, do:

i. Load batch of images and ground truth boxes.

ii. Perform forward pass through the model:

- Pass images through the model.

- Apply NMS to remove duplicate detections.

iii. Calculate loss:

- Objectness loss for true positive and false positive detections.

- Localization loss for predicted bounding boxes.

- Classification loss for predicted class labels.

- Total loss = objectness loss + localization loss + classification loss.

iv. Perform backward pass:

- Compute gradients of loss with respect to model parameters.

- Update model parameters using optimizer

Step3. Evaluate model on validation set.

Step4. Save model if it performs better than previous best.

Inference Algorithm for YOLOv7

Step1: load pre-trained model

Step 2: for each test image do

i. pass image through model

ii. apply NMS to remove duplicate detections

iii. for each detection do

if objectness score > threshold then

-draw predicted bounding box on image

-label box with predicted class and confidence score

4. Evaluation metrics

The following metrics were used in this study to assess the model's performance: precision, recall, F1 score, and mAP0.5. The percentage of accurately detected targets to all detected targets is the definition of precision, a frequently used evaluation metric. A greater Precision number usually denotes better detection performance. Although precision is a crucial evaluation parameter, it might not be enough for a thorough analysis. For a more extensive investigation, more metrics like recall and F1 score were also applied. The following formulas were used to calculate the values of Precision, Recall, and F1 score:

Precision:

$$P = \frac{\mathrm{TruePositive}}{\mathrm{TruePositive}+\mathrm{FalsePositive}} \mathrm{x}100\% \quad (1)$$

Recall:

$$R = \frac{\mathrm{TruePositive}}{\mathrm{TruePositive}+\mathrm{FalseNegative}} \mathrm{x}100\% \quad (2)$$

F1 score:

$$F1 = 2\mathrm{x}\frac{\mathrm{PxR}}{\mathrm{P}+\mathrm{R}} \quad (3)$$

mAP0.5:

The term "mAP0.5" refers to the intersection over union (IoU) metric's mean average precision at a threshold of 0.5. This metric serves as a common performance indicator for object detection models. When a model's predicted bounding boxes are compared to ground truth bounding boxes at an IoU threshold of 0.5, the mAP0.5 metric calculates the model's average precision. By definition, a prediction is considered accurate if its bounding box coincides by at least 50% with the bounding box of the ground truth.

In Eqs. (1) and (2), the term True Positive, indicates count of correctly detected projector screens; False Positive, indicates count of non-projector screens detected as projector screens and False negative, indicate projector screens count that are not detected.

5. Experimental Setup

The experiments of this study were carried out on the Google Collaboratory platform using Python 3.9, Pytorch (version 1.13.1), and CUDA (version 12) for training the YOLOv7 model. We trained the detection model using Tesla T4 (16G) GPU.

6. Experimental Results and Discussions

The results of our custom YOLOv7 model is shown in the Fig. 6, to validate the efficiency of custom YOLOv7 model we compared the performance with basic YOLOv7 and Retinanet (source from GitHub repository [37]) models trained separately with both augmented dataset and original dataset. The Table 1

shows experimental results comparison of our custom YOLOv7 model with the other models considered for evaluation.

Table 1
Comparison of custom YOLOv7 model with other models

Model	Dataset	Precision	Recall	F1 score	mAP0.5	Training Tme	Inference Time/image
CustomYOLOv7	Dataset of 5K images with augmentation	**98.6**	**96.7**	**97.64**	**99.1**	**5734 sec.**	**0.54 ms**
	Dataset of 2K images without augmentation	92	96.7	94.29	97.9	2203 sec.	0.55ms
Basic YOLOv7	Dataset of 5K images with augmentation	92.4	92	90.05	96.1	7790 sec.	0.56ms
	Dataset of 2K images without augmentation	87	94.4	90.54	95.1	2692 sec.	0.58ms
Retinanet	Dataset of 5K images with augmentation	80.3	90.4	85.05	91.4	9396 sec.	1.2ms
	Dataset of 2K images without augmentation	74.4	89.6	81.29	82.5	3276 sec.	1.3ms

From the Table 1 and Fig. 8 we can make the following observations:

1. Precision:

- CustomYOLOv7 has the highest precision on both datasets, with 98.6% on the dataset with augmentation and 92% on the dataset without augmentation.
- The basic YOLOv7 model demonstrates strong precision results, ranking second with 92.4% on the augmented dataset and 87% on the unaugmented dataset.
- The Retinanet model exhibits comparatively lower precision, registering at 80.3% on the dataset with augmentation and 74.4% on the dataset without augmentation.
- The implementation of data augmentation consistently enhances the precision performance.

2. Recall:

- CustomYOLOv7 has the highest recall on both datasets i.e., 96.7% .

- Basic YOLOv7 has the second-highest recall, with 92% on the dataset with augmentation and 94.4% on the dataset without augmentation.
- Retinanet has the lowest recall, with 90.4% on the dataset with augmentation and 89.6% on the dataset without augmentation.
- Data augmentation boosts the recall of all model.

3. F1 score:

- With an F1 score of 94.29% on the dataset without augmentation and 97.64% on the dataset with augmentation, CustomYOLOv7 has the highest overall performance.
- Basic YOLOv7 has the second-highest F1 score on both datasets, with 90.05% on the dataset with augmentation and 90.54% on the dataset without augmentation.
- The Retinanet shows a relatively lower F1 score on both datasets with 85.05% on the dataset with augmentation and 81.29% on the dataset without augmentation.
- Data augmentation improves the F1 score of all models

4. mAP0.5:

- CustomYOLOv7 has the highest mAP0.5 on both datasets, with 99.1% on the dataset with augmentation and 97.9% on the dataset without augmentation.
- Basic YOLOv7 has the second-highest mAP0.5 on both datasets, with 96.1% on the dataset with augmentation and 95.1% on the dataset without augmentation.
- Retinanet has the lowest mAP0.5 on both datasets, with 91.4% on the dataset with augmentation (indicating perfect performance) and 82.5% on the dataset without augmentation.
- Data augmentation improves the mAP0.5 of all models.

From Fig. 8 it is clear that CustomYOLOv7 outperforms the other models on all the evaluation metrics, followed by Basic YOLOv7, and then Retinanet.

The training time of CustomYOLOv7 is the shortest, i.e., 1.593 hours, compared to Basic YOLOv7 and Retinanet, for the dataset with augmentation. However, for the dataset without augmentation, the training time of Basic YOLOv7 is the shortest, i.e., 0.612 hours.

The inference time of Retinanet is the highest, i.e., 1.2 ms, compared to CustomYOLOv7 and Basic YOLOv7, for the dataset with augmentation. However, for the dataset without augmentation, the inference time of Retinanet is comparable to Basic YOLOv7.

Overall the CustomYOLOv7 outperforms Basic YOLOv7 and Retinanet in all the evaluation metrics for both datasets (with and without augmentation) and the performance of all the models is significantly improved when trained on a dataset with augmentation compared to without augmentation.

The bounding boxes generated by Retinanet is not covering whole slide region completely for many images when compared with the YOLOv7's bounding boxes(Fig. 7), which makes the Retinanet not suitable for applications which require to crop the slide region for further processing.

7. Conclusion and Future work

The projected slide region detection and extraction is an important task in Lecture video analysis of video captured in classrooms and conference rooms. An object detection process based on custom YOLOv7 model was developed, trained on the manually annotated dataset collected from various resources. We created two datasets of 2k and 5k images respectively, the 2k image dataset is collection of original images and the 5k dataset has augmented images from original 2k dataset. The custom YOLOv7 model performance is compared with Retinanet model and basic YOLOv7 model, the experimental results showed that our customYOLOv7 model has outperformed with F1 score 97.64%, Precision rate of 98.6%, Recall rate of 96.7%, and mAP of 99.1%. In addition, the models trained with augmented 5k dataset generated good results compared to 2k dataset of original images. In summary, our custom YOLOv7 model trained on 5k augmented dataset can accurately and quickly detect complete projected slide region from Lecture videos captured in class rooms/conference rooms. In the future work, we are planning develop a text-based search system on a repository of presentation videos captured in classrooms/conference rooms.

Declarations

Ethics Approval and Consent to Participate:

No participation of humans takes place in this implementation process

Human and Animal Rights:

No violation of Human and Animal Rights is involved.

Funding:

No funding is involved in this work.

Data availability statement:

Data sharing not applicable to this article as no datasets were generated or analyzed during the current study

Conflict of Interest:

Conflict of Interest is not applicable in this work.

Authorship contributions:

All authors are contributed equally to this work

Acknowledgement:

There is no acknowledgement involved in this work.

References

1. Husain, M., & Meena, S. M. (2019). Multimodal fusion of speech and text using semi-supervised LDA for indexing lecture videos. 2019 National Conference on Communications (NCC). IEEE.
2. Kanadje, M., et al. (2016). Assisted keyword indexing for lecture videos using unsupervised keyword spotting. *Pattern Recognition Letters*, *71*, 8–15.
3. Tuna, T. (2015). Automated lecture video indexing with text analysis and machine learning. *Diss*.
4. Kate, L. S., Waghmare, M. M., & Priyadarshi, A. (2015). An approach for automated video indexing and video search in large lecture video archives. 2015 international conference on pervasive computing (ICPC). IEEE.
5. Yang, H. (2011). Lecture video indexing and analysis using video ocr technology. 2011 Seventh International Conference on Signal Image Technology & Internet-Based Systems. IEEE.
6. Wangchen, T. (2022). EDUZONE–A Educational Video Summarizer and Digital Human Assistant for Effective Learning. 2022 7th International Conference on Information Technology Research (ICITR). IEEE.
7. Sun, F. (2022). and Xuedong Tian. Lecture Video Automatic Summarization System Based on DBNet and Kalman Filtering. Mathematical Problems in Engineering 2022.
8. Kota, B., Urala (2021). Automated whiteboard lecture video summarization by content region detection and representation. 2020 25th International Conference on Pattern Recognition (ICPR). IEEE.
9. Abhilash, R., Kashyap (2021). Lecture video summarization using subtitles. 2nd EAI International Conference on Big Data Innovation for Sustainable Cognitive Computing: BDCC 2019. Springer International Publishing.
10. Davila, K., et al. (2021). FCN-LectureNet: extractive summarization of whiteboard and chalkboard lecture videos. *Ieee Access : Practical Innovations, Open Solutions*, *9*, 104469–104484.
11. Lee, G. C., et al. (2017). Robust handwriting extraction and lecture video summarization. *Multimedia Tools and Applications*, *76*, 7067–7085.
12. Hassani, H., Ershadi, M. J., & Mohebi, A. (2022). LVTIA: A new method for keyphrase extraction from scientific video lectures. *Information Processing & Management*, *59*(2), 102802.
13. Medida, L., Haritha, & Ramani, K. (2021). Impact of Deep Learning on Localizing and Recognizing Handwritten Text in Lecture Videos. *International journal of Advanced Computer Science and Applications* 12.4.
14. Xu, C. (2019). Lecture2note: Automatic generation of lecture notes from slide-based educational videos. 2019 IEEE International Conference on Multimedia and Expo (ICME). IEEE.
15. Dutta, K. (2018). Localizing and recognizing text in lecture videos. 2018 16th international conference on frontiers in handwriting recognition (ICFHR). IEEE.

16. Shin, H., Valentina, et al. (2015). Visual transcripts: lecture notes from blackboard-style lecture videos. *ACM Transactions on Graphics (TOG)*, *34*(6), 1–10.
17. Yang, H., Gruenewald, F., & Meinel, C. (2012). Automated extraction of lecture outlines from lecture videos. 4th International Conference on Computer Supported Education, CSEDU.
18. Ravi, S. (2022). A Novel Educational Video Retrieval System Based on the Textual Information. Proceedings of the 13th International Conference on Soft Computing and Pattern Recognition (SoCPaR 2021). Cham: Springer International Publishing.
19. Loc, C., Vinh (2021). Content based Lecture Video Retrieval using Textual Queries: to be Smart University. 2021 13th International Conference on Knowledge and Systems Engineering (KSE). IEEE.
20. Yang, H., & Meinel, C. (2014). Content based lecture video retrieval using speech and video text information. *IEEE transactions on learning technologies*, *7*(2), 142–154.
21. Adcock, J. (2010). Talkminer: a lecture webcast search engine. Proceedings of the 18th ACM international conference on Multimedia. ACM.
22. Xu, C., et al. (2022). Semantic Navigation of PowerPoint-Based Lecture Video for AutoNote Generation. *IEEE Transactions on Learning Technologies*, *16*(1), 1–17.
23. Rahman, M., Rajiur, S., Shah, & Subhlok, J. (2020). Visual summarization of lecture video segments for enhanced navigation. 2020 IEEE International Symposium on Multimedia (ISM). IEEE.
24. Furini, M., Mirri, S., & Montangero, M. (2018). Topic-based playlist to improve video lecture accessibility. 2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC). IEEE.
25. Monserrat, T. J., Palma, K. (2013). Notevideo: Facilitating navigation of blackboard-style lecture videos. Proceedings of the SIGCHI conference on human factors in computing systems.
26. Wang, C. Y., Bochkovskiy, A., & Liao, H. Y. M. (2022). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. arXiv preprint arXiv:2207.02696.
27. Lin, T. Y. (2017). Focal loss for dense object detection. Proceedings of the IEEE international conference on computer vision.
28. Li, K., et al. (2014). Structuring lecture videos by automatic projection screen localization and analysis. *IEEE transactions on pattern analysis and machine intelligence*, *37*(6), 1233–1246.
29. Rajgure, S., Oria, V., & Gouton, P. (2014). Slide localization in video sequence by using a rapid and suitable segmentation. *marginal space. Color Imaging XIX: Displaying, Processing, Hardcopy, and Applications* (Vol. 9015). SPIE.
30. Soundes, B., Larbi, G., & Samir, Z. (2019). Pseudo Zernike moments-based approach for text detection and localisation from lecture videos. *International Journal of Computational Science and Engineering*, *19*(2), 274–283.
31. Zhao, B., et al. (2018). A novel approach to automatic detection of presentation slides in educational videos. *Neural Computing and Applications*, *29*, 1369–1382.
32. Thomas, C., et al. (2022). Automatic prediction of presentation style and student engagement from videos. *Computers and Education: Artificial Intelligence*, *3*, 100079.

33. Araujo, A. (2016). Large-scale query-by-image video retrieval using bloom filters. arXiv preprint arXiv:1604.07939.
34. Dwyer, B., & Nelson, J. (2022). Solawetz, J., et. al. Roboflow (Version 1.0) [Software]. Available from https://roboflow.com. computer vision.
35. Yang, S. (2022). Image data augmentation for deep learning: A survey. arXiv preprint arXiv:2204.08610.
36. Chien, W. (2022). YOLOv7 Repositry with all Instruction. Available online: https://github.com/WongKinYiu/yolov7 (accessed in the month of March 2023).
37. Ayushman, B., A PyTorch implementation of RetinaNet Available online: https://github.com/benihime91/pytorch_retinanet (accessed in the month of March 2023).

Figures

Figure 1

Sample frames of presentation videos

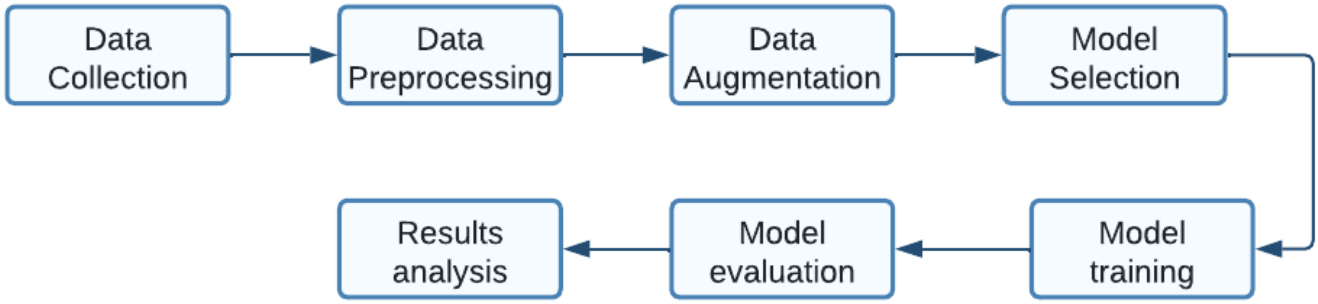

Figure 2

Work flow of object detection

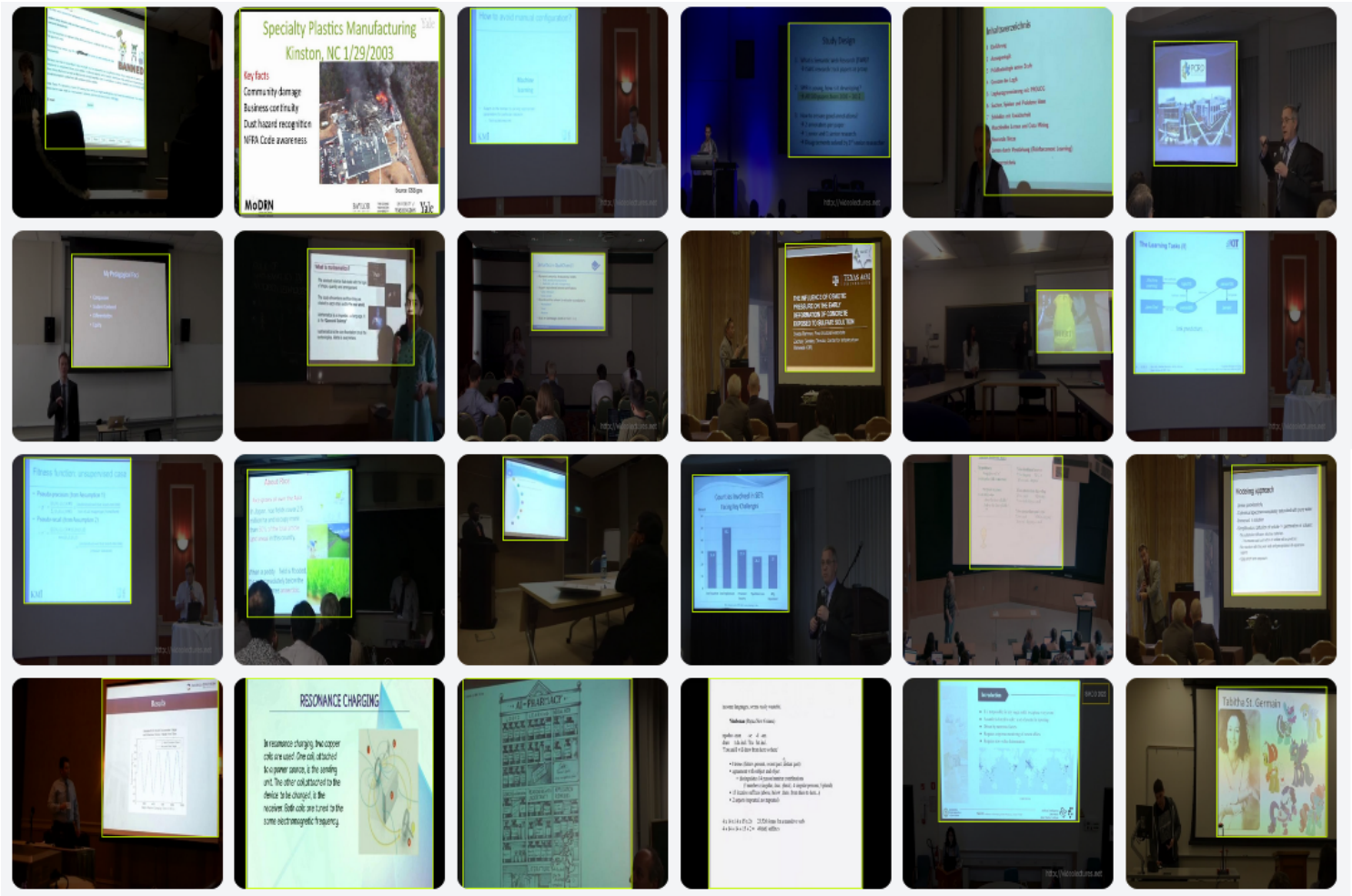

Figure 3

Dataset of Images labelled using roboflow

Figure 4

Figure 6. Test results using custom YOLOv7

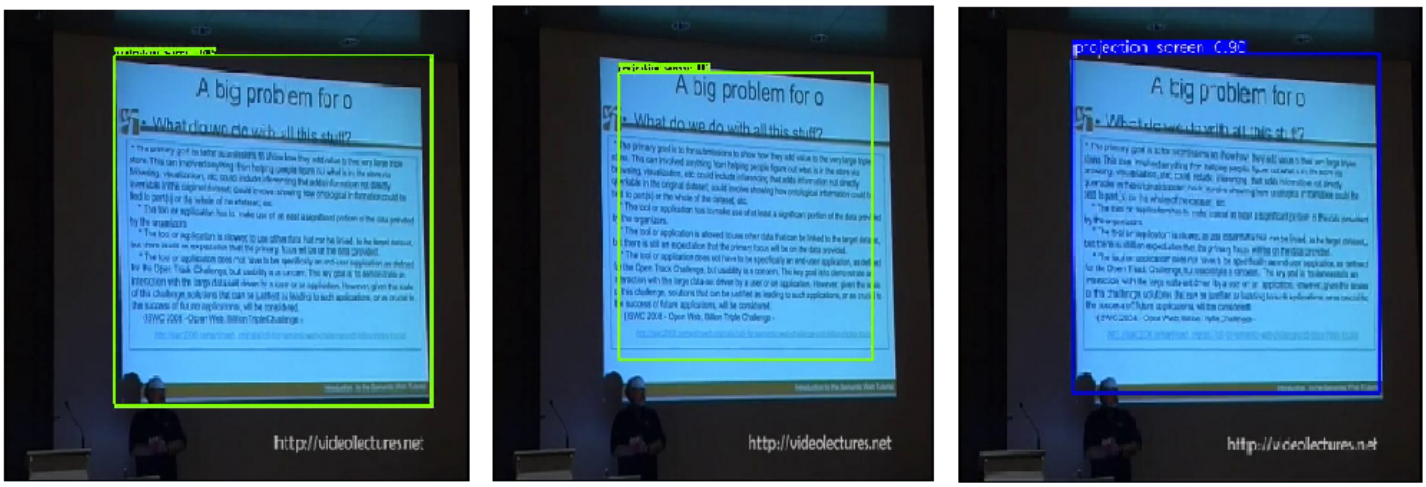

Figure 5

Figure 7. Test results comparing Retinanet and YOLOv7

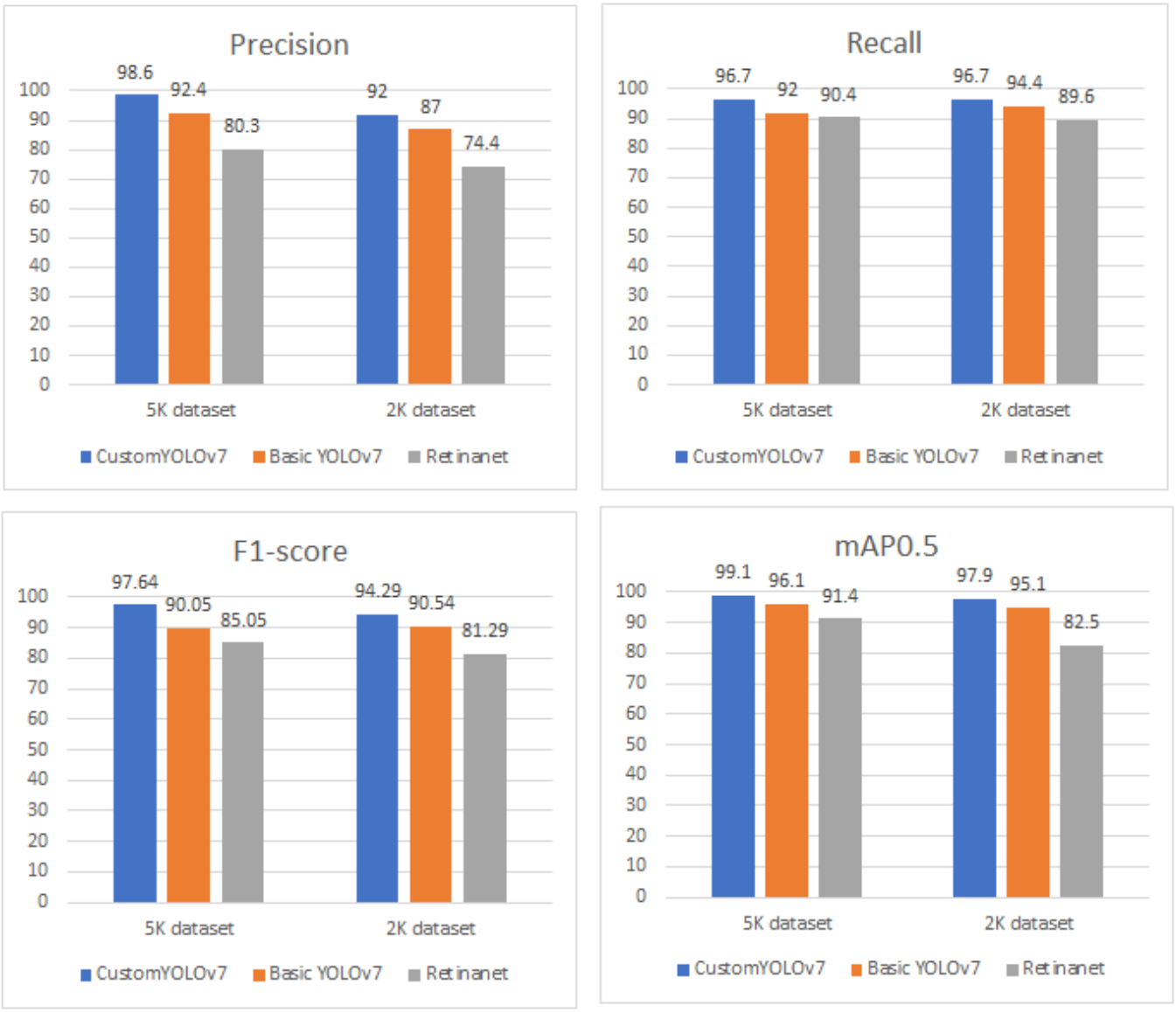

Figure 6

Figure 8. Performance comparison of custom YOLOv7, Basic YOLOv7 and Retinanet