Project LEAFS: Learning Efficiency Assessment from Footage of Students

Abdullah Enes Ergun[1] Baha Kirbasoglu[1] Can Ali Ates[1]

Abstract

Attitudes of the students in class affect lecture efficiency for both the lecturer and themselves. Keeping track of each student's attitude simultaneously during to lecture is a really hard problem. Therefore, we propose a method to detect different attitudes of students for solving this problem in this case study. We use the YOLOv5 tool that combines deep learning and computer vision techniques. For instance, Convolutional Neural Networks. We collected our dataset using web scraping, taking classroom photos by hand and using a small portion of dataset which is available online. The dataset which is collected from the multiple sources contains seven classes. These classes are separated into three positive and four negative classes. We created a semi-theoretical formula based on these classes that calculates lecture efficiency according to detected class instance counts and weights of these classes. While positive classes increase the efficiency during lecture efficiency assessment, on the other hand, negative labels decrease. Our goal is to help lecturers to improve their lectures, based on students' attitudes.

1. Introduction

Most of the time, assessing the efficiency of a lecture for students is a problem for the lecturer. The lecturer cannot be sure if the lecture is understandable or not. In this situation, the lecturer needs a simultaneous system to keep track of the students one by one based on their attitudes. Students can act different behaviours during the lecture such as playing phone, taking notes, sleeping, etc. These attitudes can have different meanings. For instance, a student who yawns is not completely distracted from lecture, can still listen the lecture without hundred percent attention. The lecturer cannot assess all of these attitudes in real-time during a lecture, so an automated system has to detect students' attitudes and evaluates the status of the student based on what the student does. Therefore, developing a system that calculates the lecture efficiency based on semi-theoretical formula that created by ourselves with giving weights to these student attitudes and then reports the calculated lecture efficiency with detected student attitudes in class both visually and written to the lecturer in real time may help both the students and the lecturer. This developed system is supported with graphical user interface, in other words GUI, that is designed by us to create user friendly environment. Lecture efficiency basically can be assessed with seven attitudes. These attitudes can be evaluated as positive and negative. While positive attitudes such as listening, taking notes and raising a hand increase lecture efficiency; on the other hand, negative attitudes such as yawning, playing with the phone, sleeping and eating or drinking something can decrease lecture efficiency. Weights of these attitudes are determined by us scaled into zero-one range. The algorithm that we use assesses the lecture efficiency and returns a basic report to a lecturer to prevent inefficient lectures. As a result of these reports, the lecturers can change their own teaching techniques or materials, and students can focus more with these improvements then change their own behaviours.

2. Related Work

Assessing student attitudes is a rare research topic that we found a few articles about. There are several articles use different techniques to detect behaviour of students. Article[1] collects their own data from classroom videos like we made. After that, combines the temporal and action detection to create recognition model. Then, uses this recognition model for task recognition. It uses the recognition results by giving these results to different type of video captioning models such as, HACA and RecNet. This study had some troubles like ours such as misclassifications and lack of dataset. For instance, model detects reading book as playing with phone or vice versa. So, some of the misclassification occurs because of the perspective. This study provides us a perspective to detect student behaviours and misclassifications in different way, so this article can use for the future directions of our model. Article[2,4] is about detection systems to detect mobile phone usage of a person with using classical Convolutional Neural Network and Faster Region Based Convolutional Neural Network. Article[2] uses mAP@0.5 as the evaluation metric. This studies, give us an approach about detection of behaviours which can be identified by pairing another tool such as taking note detection with pencil and notebook, playing phone detection with phone, eating or drinking detection with foods and beverages. Also, explains the how can interpret the model performance over mAP@0.5 metric. The article[3] uses YOLOv5 for detection and shows us how to use dataset efficiently with image enhancement such as boost brightness and Gaussian Noise and horizontal flip as data augmentation. This is the article we mostly focus on for our project and proves the superiority of YOLOv5 over the other CNN architectures by giving detection results. In this article they used listening, looking down, lying down, standing as labels. Their main purpose is detecting student behaviours in classroom. We use YOLOv5 to detect behaviours in classroom as well. It is similar to ours. Therefore, we choose this article as our main resource. Article[5] explains a smart lab system, that detects the students equipment's to ensure lab safety and uses a smart monitoring system. In this article, they used their own dataset and YOLOv5, YOLOv7 models. This article provides an approach to develop a graphical user interface for our project to create an user-friendly environment. Article[6] is the most related work from all of these articles, it covers more class labels which are based on two different attention types as 'high attention' and 'low attention' than our project such as 'laughing', 'bored', 'focused' etc. This work focuses on facial expressions, unlike our study. The common thing between this project and our study is using YOLOv5 and evaluate the model with mAP@0.5. The differences between projects are multiple detection on one person and trying to detect more class instances. The study which covered in the related work also keeps attendance of students in CSV file based on the their detections.

3. The Approach

3.1. Dataset

We created our dataset from three different sources. The first source is Google Images, we developed a python code that we used for web scraping from this source. As a result of this scraping process, we got 7000+ images with duplicate and huge noise for instance; comics, quote, stock images which contains watermarks and different environments such as study cafes, meetings, etc. While comics and quotes are directly deleted, watermarks are evaluated as to which can be handled or not by YOLO. The watermarks which decided as cannot to be handled by YOLO were cleaned with external tools. The different environments other than classrooms are evaluated by deciding with a domain knowledge which can be beneficial or not for our model. The environments which are not beneficial are deleted. The second source is a Ready-Made dataset which is available on GitHub for an open source project[7]. This dataset contains just listening images because of that, we took small portion from this dataset. The other reason of that we took small portion is avoid the overfitting. The third and final source is that we collected by hand from our classmates in the classroom environment. The dataset that we combined from multiple sources contains seven classes. These classes are separated into two different groups negative classes and positive classes. The positive classes are listening, raising hand and taking note. The negative classes are yawning, sleeping, eat or drinking and playing with phone.

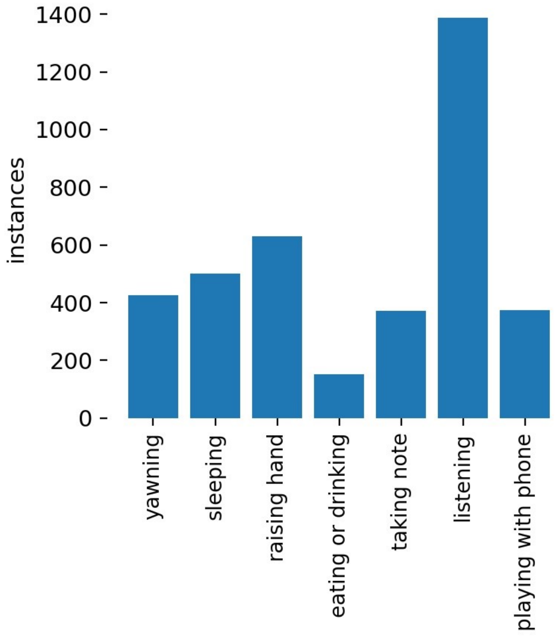


Figure 1. Distribution of Class Labels

The dataset contains imbalanced data. As an investigation over the Figure 1, listening has more instances than the other classes. This superiority of the listening class creates a high bias for our model while detection process.

3.2. YOLOv5

You Only Look Once (YOLO) is a Convolutional Neural Network based tool developed by distributed developers. In this study, we used YOLOv5 from the other YOLO versions.

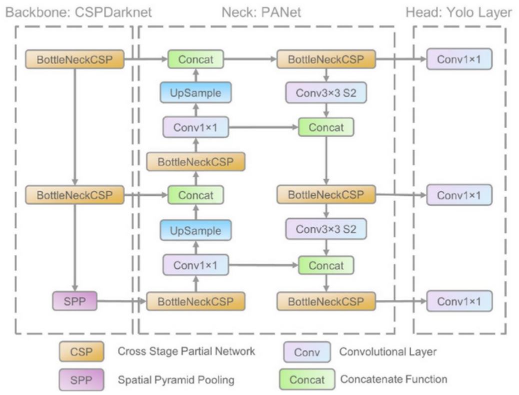


Figure 2. YOLOv5 Architecture[10]

As shown in Figure 2, YOLOv5 architecture build up from 3 different sections. Backbone is the first section which uses a convolutional neural network architecture to aggregate and form the features of images on different granularities. The second section is Neck which contains layers series to combine and mix the features of images to route them into prediction. Head is the final section which consumes the features between the routed feature from the neck, then creates the bounding boxes and makes predictions over the classes.

The architecture activates with two different activation functions which are LeakyReLU and Sigmoid. It uses LeakyReLU between convolutional layers and Sigmoid for the last fully connected layer. As an optimizer, it uses Stochastic Gradient Descent, in other words SGD, with 0.01 learning rate. It computes three different losses with two objective function. The first objective function is Binary Cross Entropy that is used for calculation of classes and objectness losses. The second objective function is Complete Intersection over Union which is used to calculate location loss. The general loss function is a combination of these losses with their weights.

$$Loss = \lambda_1 L_{cls} + \lambda_2 L_{obj} + \lambda_3 L_{loc}$$

Figure 3. Formula of Loss Function[10]

YOLOv5 is an algorithm that uses a grid system which means inputs are split into grid before feature extraction for object detection. Object detection is a computer vision and image processing technology that focuses on detection of objects in images, videos, or real time footage. YOLOv5 uses bounding box technique to detect object. In other words, it detects objects and their coordinates in images or videos simultaneously.

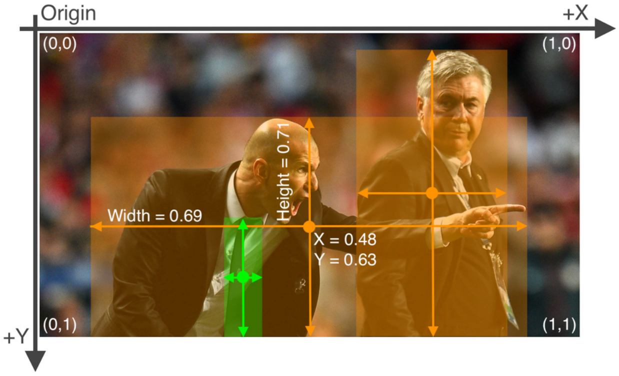


Figure 4. YOLOv5 Bounding Box Method[9]

While the older versions of YOLO are using the DarkNet architecture, YOLOv5 is the first YOLO version that uses PyTorch framework at the background. YOLOv5 has five different models such as n, s, m, l and x. As you can see in Figure 5, these models has different accuracy and performance.

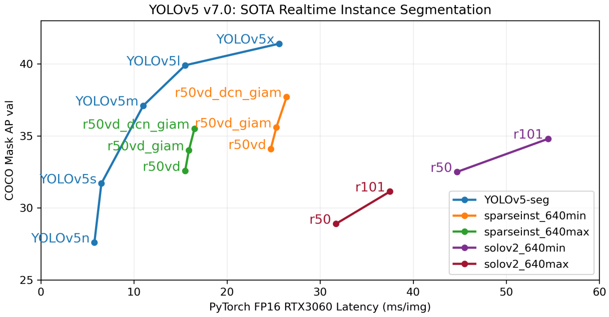
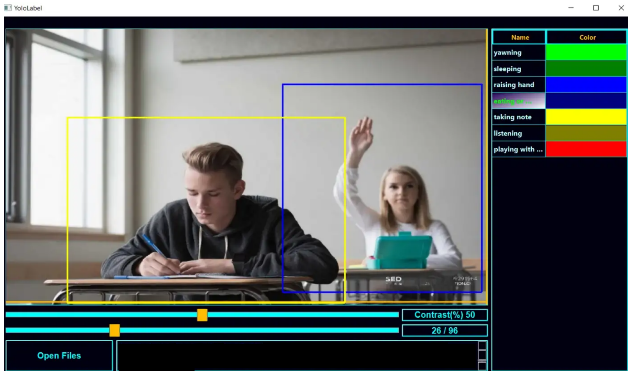


Figure 5. YOLOv5 Models[8]

As shown in Figure 5, the YOLOv5s model is an optimal model for our study. The reason behind this decision is it has a sufficient average precision value when compared with the others based on consumed time. YOLOv5s is already taking too much time to train. Because of time consuming on training process, we decided to use YOLOv5s.

3.3. Labelling Process

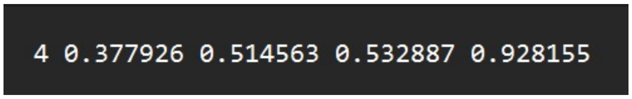
Labelling is a procedure before the YOLOv5 model training stage. We labelled train and validation images with using the YoloLabel tool shown in Figure 6. The first step of this tool is, uploading the data source and then also uploading the text file which contains names of classes shown below in the Figure 6. After these steps, we labelled the images based on classes.



Name	Color
yawning	🟩
sleeping	🟩
raising hand	🟦
eating or ...	🟦
taking note	🟨
listening	🟫
playing with ...	🟥

Figure 6. YoloLabel Tool

When images are labeled, the tool creates an annotation file which is a text file as shown in Figure 7.



```
4 0.377926 0.514563 0.532887 0.928155
```

Figure 7. Annotation Text File

The first column is the class number which is enumerated according to a certain label text file. The second and third columns contain the x and y coordinate of the bounding box center respectively. The fourth and fifth columns contain the width and height values of the bounding box respectively. These values are demonstrated in Figure 4.

3.4. Evaluation Metric

YOLOv5 uses mAP@0.5 value which stands for mean Average Precision at 0.5 threshold. Intersection over Union is a metric to check trueness of object classification.

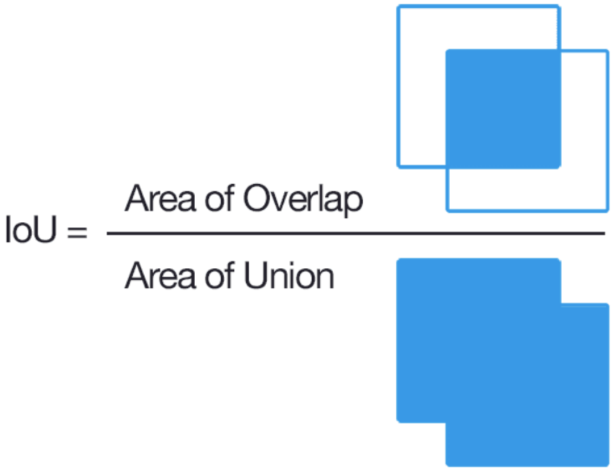


Figure 8. Calculation of Intersection over Union

If IoU value is bigger than or equal to the threshold which is 0.5, the object is classified as true. Otherwise, it is classified as false.

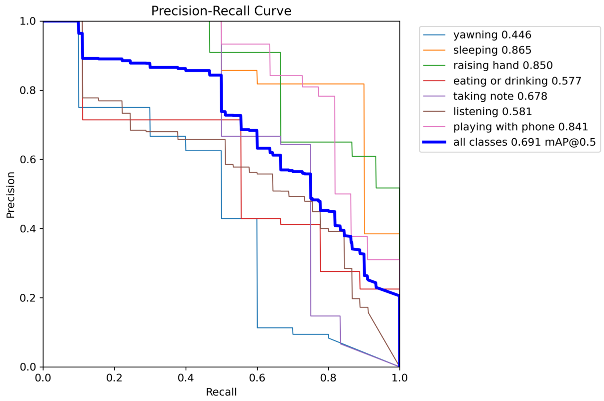


Figure 9. First Model's PR Graph

As shown in Figure 9, each class has its own Average Precision value which is equal to the area under of their own PR curve. Also, a general mAP value is calculated with taking mean of all classes average precision values. The model evaluated with this mAP@0.5 value. According to this evaluation, sleeping has the highest AP value as 0.865 and yawning has the lowest AP value as 0.446.

3.5. Lecture Efficiency Formula

We have developed a semi-theoretical formula to evaluate the effectiveness of a lecture. This formula assigns weights to different behaviors exhibited by students during the lecture. Positive behaviors are given a weight of 1, while negative behaviors are given a weight between 0 and 0.75. For example, sleeping is assigned a weight of 0, yawning is assigned a weight of 0.50, eating or drinking is assigned a weight of 0.75, and playing with a phone is assigned a weight of 0.75. We chose these weights because we believe that a student who yawns or eats or drinks is still paying attention to the lecture, but is somewhat distracted. On the other hand, a student who sleeps or plays with a phone is fully distracted and not paying attention to the lecture. Figure 10 illustrates how the efficiency of the lecture is calculated using our semi-theoretical formula.

$$\text{Lecture Efficiency} = \frac{\sum_{i=1}^{n} w_i * c_i}{\sum_{i=1}^{n} c_i} * 100\%$$

n: Number of classes

w: Determined weight of a class

c: Detected instance count of a class

Figure 10. Semi-theoretical Lecture Efficiency Formula

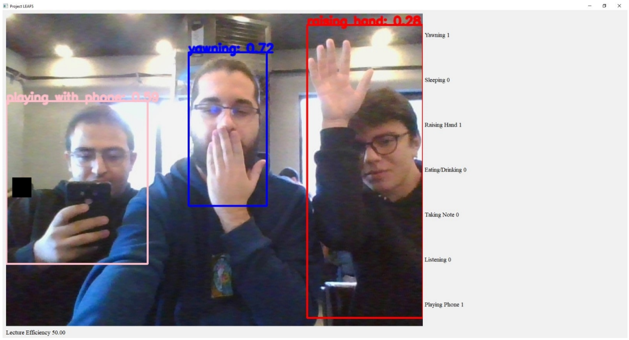
For example, as shown in Figure 11, if we have 4 people in our class and they are sleeping, listening, raising their hand, and yawning, we can use our formula to calculate the effectiveness of the lecture.

$$\frac{0*1+1*1+1*1+0.5*1}{1+1+1+1} * 100\% = 62.5\%$$

Figure 11. Sample Calculation

3.6. Graphical User Interface

As a final step in this study, we designed a Graphical User Interface (GUI) to display the detection results of the model and calculated lecture efficiency from these detections. The GUI includes real-time camera footage with detections, the number of instances of each behavior detected in real-time, and a calculation of the semi-theoretical lecture efficiency based on the behaviors detected. The GUI provides a user-friendly environment with these properties. This makes our project interpretable. As shown in Figure 12, the frame that is processed by the model shown in upper left part, counts of detected class instances shown in right part and lecture efficiency at the bottom left part of the GUI.



Yawning 1

Sleeping 0

Raising Hand 1

Eating/Drinking 0

Taking Note 0

Listening 0

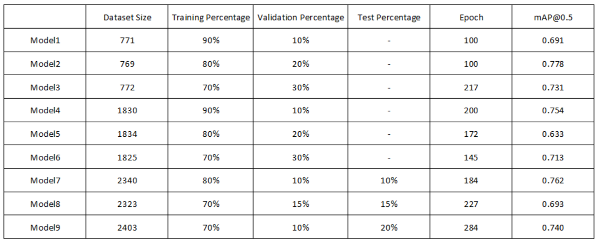
Playing Phone 1

Lecture Efficiency 50.00

Figure 12. Graphical User Interface

4. Experimental Results

4.1. Model Selection



	Dataset Size	Training Percentage	Validation Percentage	Test Percentage	Epoch	mAP@0.5
Model1	771	90%	10%	-	100	0.691
Model2	769	80%	20%	-	100	0.778
Model3	772	70%	30%	-	217	0.731
Model4	1830	90%	10%	-	200	0.754
Model5	1834	80%	20%	-	172	0.633
Model6	1825	70%	30%	-	145	0.713
Model7	2340	80%	10%	10%	184	0.762
Model8	2323	70%	15%	15%	227	0.693
Model9	2403	70%	10%	20%	284	0.740

Figure 13. All Trained Models

We trained nine models with different epochs and train-valid-test percentage as shown in Figure 13. We chose the seventh model from nine models. The seventh model has the highest mAP@0.5 value among all mAP@0.5 values and in real-time testings it detects the classes more accurately than the others. We did not separate data for testing until seventh model because we tested the model with real-time footage from webcam and YouTube videos. At the seventh, eighth and ninth model we used a test set due to increase in the dataset size and get better models.

4.2. Model

As we mentioned before, we used YOLOv5 model for this experiment. We trained our model with using our labelled dataset which contains 1892 images for training, 224 images for validation and 224 images for testing. There are several parameters for training process such as, image size, batch size, number of epochs. Image size is important factor for training. Increasing the image size usually leads to better results, but takes longer time to process. It is a trade-off between time and accuracy. We chose 640 pixels for image size to train our model. It is enough to detect attitudes correctly. The second attribute is batch size. Batch size controls the merging mechanism of images. Batch size directly affects the training time. Higher batch size means it combines more images to give a model so training time become shorter. In other words, batch size and training time are inversely proportional. We chose 64 as batch size. The last parameter is number of epochs. More epochs make model better, because it learns more and more. However, it can overfit as well. Because of overfitting, sometimes reducing the number of epochs can help. There are several ways to avoid overfitting such as increasing the dataset size or hyper-parameter tuning with the validation set. Therefore, we used a validation set to avoid possible overfitting situations. We chose 300 epochs for our model but it early stopped at 184th epoch. The reason of this early stopping is the model stops when it did not develop itself for the last 100 epochs.

4.3. Results

As mentioned in the model part, we obtained our model with parameters which are 640x480 image size, 64 batch size and 184 epochs.

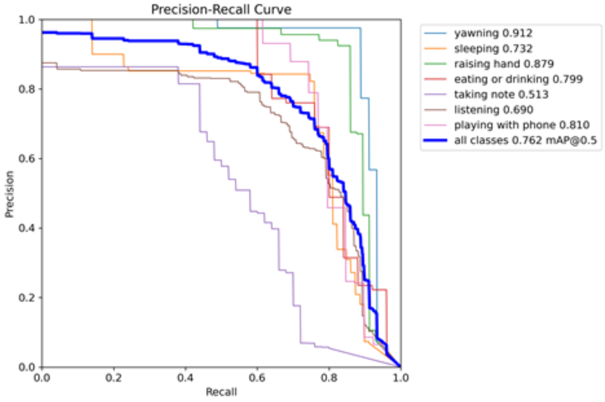


Figure 14. Model's PR Curve

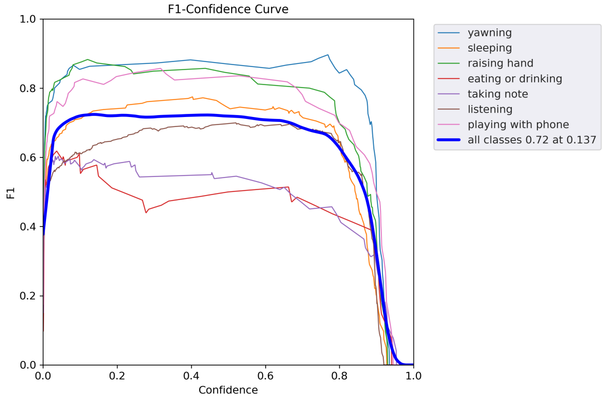


Figure 15. Model's F1 Curve

In Figure 14, the Area Under Precision-Recall Curve is a measure used to evaluate the performance of the model. The Precision-Recall Curve shows that the model performs better than average for the behaviors of sleeping, raising a hand, and playing with a phone. However, the model performs worse than average for the behaviors of eating or drinking, listening, and yawning. The performance for the behavior of taking notes is average compared to the other behaviors.

In Figure 15, the F1 Score-Confidence graph displays the F1 score value for a given confidence interval. Classes with a higher F1 score for a larger confidence interval can be predicted with greater accuracy than those with a lower F1 score value.

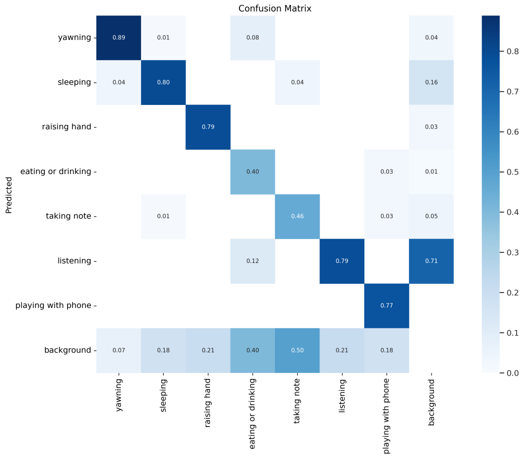


Figure 16. Model’s Confusion Matrix

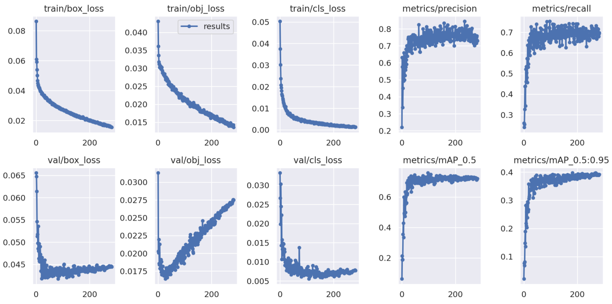


Figure 17. Model’s Result Curves

A confusion matrix is a table that is often used to define the performance of a classification algorithm. In the case of YOLOv5, a confusion matrix would show the number of true positives, false positives, true negatives, and false negatives that the model has produced. The confusion matrix is usually used to compute some metrics such as precision, recall, f1-score, and accuracy. The confusion matrix of YOLOv5 can be used to evaluate the performance of the model on the task of object detection, and fine-tune the hyper-parameters if needed. In Figure 16, background shows the False Positive on vertical scale and False Negatives are horizontal scale. According to these metrics, the False Positive of listening class is too high so it effects to model badly. Also, the False Negative of taking note class is too high that means it cannot find taking note accurately or it detects taking note as another class. In Figure 17, the graphs show losses, precision, recall and mAP@0.5 values for each epochs.

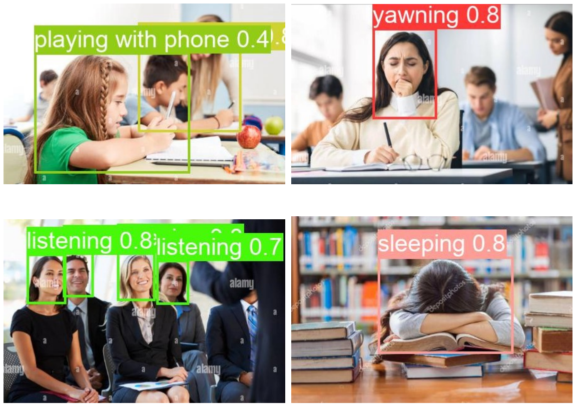


Figure 18. Testing Results

Figure 18, shows our testing results of model. As shown on the images, our model can be indecisive depending on the perspective because of the lack of data. In first image the confidence score of the model is a bit low. The reason of this low score is model gets confused about phone and pen so, it detects ’playing with phone’ as ’taking note’ and vice versa. As a result, we tried our model in real-time with 720p, 30 fps camera and show the detections, lecture efficiency and detected class instances on GUI to make it user friendly.

Conclusion

In conclusion, the proposed method for detecting the attitudes of students in a lecture and using those attitudes to assess lecture efficiency represents a significant advancement in the field of education. The use of the YOLOv5 tool and deep learning techniques, along with a semi-theoretical formula based on positive and negative classes, allows the system to accurately and efficiently identify a range of student attitudes in real-time. This information can then be used to provide a report to the lecturer on the overall efficiency of the lecture, enabling them to make adjustments to their teaching techniques or materials as needed to improve the effectiveness of the lecture and the learning experience for students.

One of the major advantages of the proposed method is its user-friendliness. The use of a graphical user interface makes it easy for lecturers to use and interpret the results of the assessment, allowing them to quickly and easily identify areas for improvement and make necessary changes. This is particularly important in the fast-paced environment of lectures, where real-time feedback is crucial for making timely adjustments.

However, it is important to recognize that the proposed method does have some limitations. For example, there is the potential for misclassifications to occur due to perspective, which could impact the accuracy of the assessment. However, these limitations can be addressed through further refinement and improvement of the system, such as by expanding the size and diversity of the dataset used for training.

Overall, the proposed method has the potential to be a valuable resource for both lecturers and students. By providing real-time feedback on the effectiveness of lectures and offering a clear path for improvement, it can help to enhance the learning experience for students and the teaching experience for lecturers. This is especially important in today's educational landscape, where the use of technology is increasingly prevalent and there is a need for effective and efficient methods for assessing and improving the quality of lectures.

Possible directions for future work include expanding the dataset used for training in order to improve the accuracy and robustness of the system, as well as incorporating additional features and capabilities. For example, the system could be enhanced to allow for the detection of more subtle or nuanced student attitudes, such as confusion or frustration. Additionally, the system could be integrated with other educational technologies, such as learning management systems or virtual reality environments, to provide a more comprehensive and seamless learning experience for students. Finally, further research could be conducted to evaluate the effectiveness of the system in different learning contexts and settings, and to identify best practices for implementing and utilizing the system in real-world educational environments.

References

[1] Bo Sun, Yong Wu, Kaijie Zhao, Jun He, Lejun Yu, Huanqing Yan and Ao Luo. Student Class Behavior Dataset: a video dataset for recognizing, detecting, and captioning students' behaviors in classroom scenes *SpringerLink* https://link.springer.com/article/10.1007/s00521-020-05587-y

[2] Rajput, Poonam Nag, Subhrajit Mittal, Sparsh. (2020). Detecting Usage of Mobile Phones using Deep Learning Technique. https://www.researchgate.net/publication/343290099

[3] Tang, Longyu Xie, Tao Yang, Yunong Wang, Hong. (2022). Classroom Behavior Detection Based on Improved YOLOv5 Algorithm Combining Multi-Scale Feature Fusion and Attention Mechanism. *Applied Sciences*. https://www.researchgate.net/publication/361841996

[4] Architha Ramesh, 2020, Monitoring Mobile usage in Classroom, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH TECHNOLOGY (IJERT) NCAIT – 2020 (Volume 8 – Issue 15), https://www.ijert.org/monitoring-mobile-usage-in-classroom

[5] Ali, L.; Alnajjar, F.; Parambil, M.M.A.; Younes, M.I.; Abdelhalim, Z.I.; Aljassmi, H. Development of YOLOv5-Based Real-Time Smart Monitoring System for Increasing Lab Safety Awareness in Educational Institutions. Sensors 2022, 22, 8820. https://doi.org/10.3390/s22228820 https://www.mdpi.com/1424-8220/22/22/8820

[6] M. M. A. Parambil, L. Ali, F. Alnajjar and M. Gochoo, "Smart Classroom: A Deep Learning Approach towards Attention Assessment through Class Behavior Detection," 2022 Advances in Science and Engineering Technology International Conferences (ASET), 2022, pp. 1-6, doi: 10.1109/ASET53988.2022.9735018. https://ieeexplore.ieee.org/abstract/document/9735018/

[7] https://github.com/it-maranatha/classroom$_{dataset}$

[8] https://github.com/ultralytics/yolov5

[9] https://github.com/ultralytics/yolov5/discussions/7370

[10] https://iq.opengenus.org/yolov5/

[11] https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/