Indonesian Journal of Electrical Engineering and Informatics (IJEEI)
Vol. 8, No. 4, December 2020, pp. 757~769
ISSN: 2089-3272, DOI: 10.11591/ijeei.v8i4.2585

Student Activity Detection Using Deep Learning with YOLOv3

Md. Yousuf Ali[1], Xuan-De Zhang[2], Md. Harun-Ar-Rashid[3]
[1,2]Department of Computer Science and Technology, Shaanxi University of Science and Technology, Xian, China
[3]Department of Computer Science and Engineering, MawlanaBhashani Science and Technology University, Tangail, Bangladesh

Article Info

Article history:

Received Jul 2, 2020
Revised Dec 2, 2020
Accepted Dec 15, 2020

Keyword:

YOLOV3,
PASCAL VOC2012,
Faster R-CNN.

ABSTRACT

This article describes the main phases of a new learning system by YOLOv3 is used for deep learning to identify student activities. Any unwanted problems in SUST- (Shaanxi University of Science and Technology) can be circumvented by using this process. In this article, we have investigated the problem of image-based student activity detection in SUST. It involves making a prediction by analyzing student poses, behavior, and activities with objects from complex images instead of videos. Comparing with all approaches, we conclusively decided to use an algorithm YOLOv3 (You Only Look Once) which is the latest and more convenient. The algorithm utilizes anchor boxes, bounding boxes, and a variant of Darknet. We have created our own dataset collecting images from SUST and annotated the dataset manually. During the research with this project, we have considered student activities in the SUST into seven sections namely reading, phoning, using a laptop, taking books, smiling, looking, and sleeping. The proposed system provides not only multi-tasking knowledge with classification but also localization of students and the equivalent actions instantaneously. Our intention is to detect the student position automatically, efficiently, confidently, and strictly with the help of extracted image functions. Interestingly, the proposed approach achieved a mean average precision (mAP) of 97%. In the future, a combination of real-time data analysis will improve the value of this scheme.

Corresponding Author:

Md. Yousuf Ali,
Department of Computer Science and Technology,
Shaanxi University of Science and Technology,
Xian, China,
Email: mhr.rashid.bd@gmail.com

1. INTRODUCTION

Computer vision and pattern recognition is a vast area in recent research field. From the sector of computer vision we attain knowledge from digital images and videos. In this arena, a vital topic is student activity detection. Activity recognition is an essential expertise in extensive computing as it can be applied to many real-life student-centric problems. In the library, students are not only confined to study but also many other activities. Sometimes, they create an unavoidable noisy atmosphere which is irritating for others as well as authorities. Student behaviors and activity in the library often demand to be monitored for preservation reasons and other purposes. In our article, we focused on their various activities and detect these actions through images instead. We have considered few activities and categorized those into five sections like reading, phoning, sleeping, taking the books, using the laptop. We have tried to outline a solution for recognition of student behaviors in the library using YOLO [1] (You Only Look Once) and faster R-CNN [2] approach for activity detection in still image based. Students glimpse at an image and know instantly what activities are done by the people in the library. The system YOLO trains on full pictures and adjusts activities detection directly. We have used third and latest version of YOLO that is YOLOv3. It's a little conspicuous than the last

versions but more accurate and fast. YOLOv3 calculates an object's grade for each bounding box which uses logistic regression. Each of the box predicts the classes and for the class projections, we have used binary cross-entropy loss while training. For outstanding performance, we used independent logistic classifiers instead of softmax because it is worthless.

Our study provides an outline to computer vision including fundamentals of image realization, feature detection and matching, and also classification. Though a lot of works have been introduced in this sector, we focused only activity detection which is the recent topmost [3], [4]. What activity students are performing is identifying only from an image instead. In our daily life, many unethical, unsocial activities are occurring everywhere. Various strict and innovative systems are also being developed for halting them. So, it will be very progressive, if there exists an artificial system where student activities are being recognized automatically from images. That's why we choose this topic. Although initially, we work only on still images, the same system may be applied to video too. Adding more types of activity will add more variety. The further improvement as taking a snap of rules breaking activity will make it more useful

For summarization, the main contributions can be represented as

1) How an effective object detection algorithm can be applied to detect action.
2) Created PASCAL VOC 2012 dataset called 'SUST-S-Act' containing 150 image data.
3) Utilized image data that is computationally less expensive and achieved a satisfactory result.

1.1. Literature review and Related Works

Student activities detection and localization is an ongoing research topic in computer vision. A lot of approaches have been provided in the last two decades. But surprisingly most of them on video or sequential images as well as at the initial stage those were only for action recognition. At first, we will discuss those methods.

1.2 Activity detection in still images and videos

We can broadly classify the existing methods into this action recognition and detection. We can say that they have some detected on pose based, context-based, and Part-based methods. There exist very few works of specific human activities detection in still images. Human action, [5].An approach to pose based action recognition. [6], Learning person-object interactions for action recognition in still images. Contextual action recognition with R*CNN. J. Sung [7], apply human activity detection from RGB-d images," in AAAI Workshop on Pattern, Activity and Intent Recognition. Learning human activities [8], and object affordances from the RGB-d image. Learning context for collective activity recognition. Recognizing human activities from partially observed videos and images [9]. Recognizing Actions through Action-Specific Person Detection" IEEE transactions on image processing. Discriminative order let mining for real-time recognition of human-object interaction. Activity net: A large-scale image benchmark for human activity understanding. Sequential deep learning for human action recognition [10], In Proceedings of the Second International Conference on Human Behavior Understanding. An approach to pose based action recognition. Simonyan, Karen & Zisserman Stream Convolutional Networks for Action Recognition in Videos" Advances in Neural Information Processing [11]. They explicitly created motion features in the form of accumulated optical flow vectors. That is why instead of using a single network for spatial context, this methodology has two separate networks - one for spatial context (pre-trained) and another one for motion context.

2. RESEARCH METHODOLOGY

2.1. YOLOv3 and Architecture

The entire image is passed on only once over the network. SSD is an additional platform for object finding algorithms that forward the image once through a deep learning network. However, YOLOv3 [12], overflows faster than SSD, achieving an equivalent level of accuracy. We can think of activity recognition as a combination of an object finder and object recognizer. In old computer vision approaches, a window was used to search for objects in very different places and scales. As a result, it was assumed from such an upscale operation that the ratio of the thing was typically assembled. Early deep learning-based algorithm rules for object detection, such as R-CNN and Fast-R-CNN [13], used a technique known as a selective search to reduce the number of bounding frames the algorithm had to check. We basically run our neural network on new images at experiment time to predict detections. Our core network runs at 45 structures per second by means of no batch processing on a Titan X GPU and a fast version runs at more than 150 fps. With these funds, we can develop a streaming video in real-time [14], with less than 25 milliseconds of invisibility. Additionally, YOLO accomplishes more than twice the mean average accuracy of other real-time systems. We followed this site for our system runs please check out this link http://pjreddie.com/yolo/.

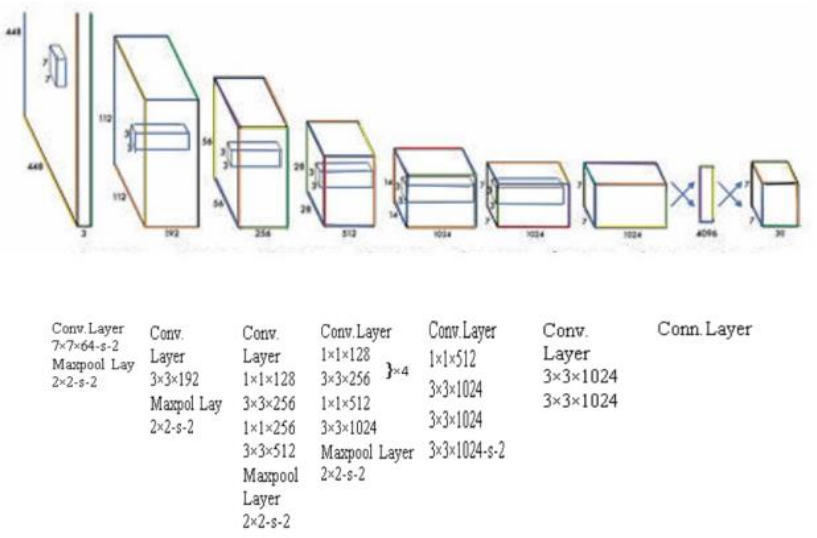

Figure 1. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1 × 1 convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224 × 224 input image) and then double the resolution for detection

This architecture we follow for our activity detection task. Dividing each image into S x S regions and within each region, it directly sinks to find B bounding boxes and a score for each of the C classes. It is the key idea of YOLO. For each of the B bounding boxes, there are center x, center y, width, height, and confidence of the bounding box. There will only be one set of class scores C for all bounding boxes in that region. The output of the YOLO network will be a vector of S× S × (5B + C) numbers for each image. YOLO was pertained on Image Net with S = 7, B = 2, and C = 20. In general, the existing YOLO architecture consists of 24 convolution layers followed by 2 connected layers and a final output layer. Since there are only 5 classes of actions, our last layer requires C = 5. The final edition of our network is a prediction of 7 × 7 × 30 tensors.

2.1.1 YOLOv3 detects the objects in prearranged image

Firstly, it divides the image into S×S and the estimated grid of cells. The scale of these 169 cells varies calculation on the scale of the input. For a 448×448 input size that we have determined to utilize in our experiments, the cell size was 32×32. Each cells are charged separately in all boxes for the prediction of phase picture. For each bounding box, the network also predicts the confidence that the bounding box encloses an object and the probability of the enclosed object being a particular class. Most of these bounding boxes are eliminated because their confidence is low, or they are enclosing the same object as another bounding box with a very high confidence score. YOLO v3 handles multiple scales. They have also improved the network by making it bigger and residual networks by adding shortcut connections.

2.1.2 YOLO loss function and restriction

The loss function can be divided into five sections, in which sections (i) and (ii) are focusing on the loss of the bounding box coordinates, sections (iii) and (iv) are scolding the differences in the confidence of having an object in the grid and section (v) is scolding for the difference in class probability. The loss function for the bounding box size is based on the square root of the dimensions, which is an interesting part to note. The small deviations in longer bounding boxes should provoke less of a penalty than in miniature bounding boxes. The "lamda-coord" hyper-parameter is set to assure “fair” contribution of the bounding box location penalty and the classification penalty to the overall loss function. The "lamda-noobj" is set to scold less for the confidence of identifying an object when there is not one.

$$\text{Loss} = \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{\text{obj}} \left(x_i - \check{x}_i\right)^2 + \left(y_i - \check{y}_i\right)^2 \quad (1)$$

$$+\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{\text{obj}} \left(\sqrt{w_i} - \sqrt{\check{w}_i}\right)^2 + \left(\sqrt{h_i} - \sqrt{\check{h}_i}\right)^2 \quad (2)$$

$$+\sum_{i=0}^{S^2}\sum_{j=0}^{B} 1_{ij}^{obj}\left(C_i-\check{C}_i\right)^2 \quad (3)$$

$$+\lambda_{noobj}\sum_{i=0}^{S^2}\sum_{j=0}^{B} 1_{ij}^{noobj}\left(C_i-\check{C}_i\right)^2 \quad (4)$$

$$+\sum_{i=0}^{S^2} 1_{ij}^{obj}\sum_{CEClasses}\left(p_i(c)-\check{p}_i(c)\right)^2$$

Because each grid cell predicts only two fields and can have only one category, YOLO places severe spatial constraints on predicting bounding boxes. This spatial limitation limits the number of nearby objects that our model can predict. Our model fights small groups of items like flocks of birds. Because our model learns to predict bounding boxes from data, it is difficult to generalize to objects with new or unusual aspect ratios or configurations. Our model also uses relatively rough features to predict the bounding box because our architecture contains multiple levels of down sampling from the input image. Although our training loss function can approximate recognition performance, the loss function handles errors when dealing with small and large bounding boxes in the same way. Small errors in large boxes are usually harmless, but small errors in small boxes have a greater impact on the IOU. The main source of error is error localization.

2.2. YOLO-V3 Complete Training Diagram

Understanding motion from still images is not an easy task. With the presence of motion, it is far easy to detect action. We need to estimate the place and pose of the person in still images. Whenever it attains to the faster object detection algorithm, we all think about YOLO. However, it has some accuracy arguments. In our work, we adopted YOLOv3 which is better, more accurate, and a little slower than YOLOv2. The YOLO v3 manages the more complex architecture of Darknet [15], which makes it slower but develops its accuracy. YOLO v3 has provided us a 106 layer fully convolution architecture. It applies a variant of Darknet. It makes the detection in three different scales which is the most conspicuous feature of v3. The input image dimensions are 32, 16, and 8 sequentially. YOLO v3 uses 9 anchor boxes. We used K-Means clustering to generate 9 anchors. YOLO v3 predicted more extended bounding boxes than YOLO v2. It might be performed multi-label classification for objects detected in images.

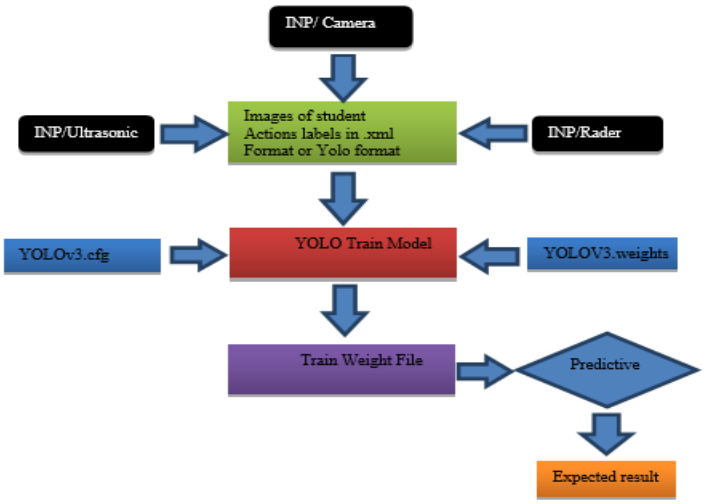

Figure 2. YOLO-V3 complete training diagram

First, we transformed the "SUST_S_Act" dataset in the form of a YOLOV3 supported format. Then we added some files to the YOLO training model. Which we specify the number of actions and their names, mention the path where the train weight file will be saved, mention the configuration file, which contains all layers of YOLO algorithm, pre-trained convolution weights then the predictive condition of images after that we got expected result.

2.3. Faster R-CNN Network Design

To obtain accurate object recognition results, a large number of proposed regions for the fast R-CNN [9] generally have to be generated in the selective search. Faster R-CNN replaces the selective search with a

region suggestion network. This reduces the number of proposed regions and at the same time ensures precise object activity recognition. We have shown in Figure 3.

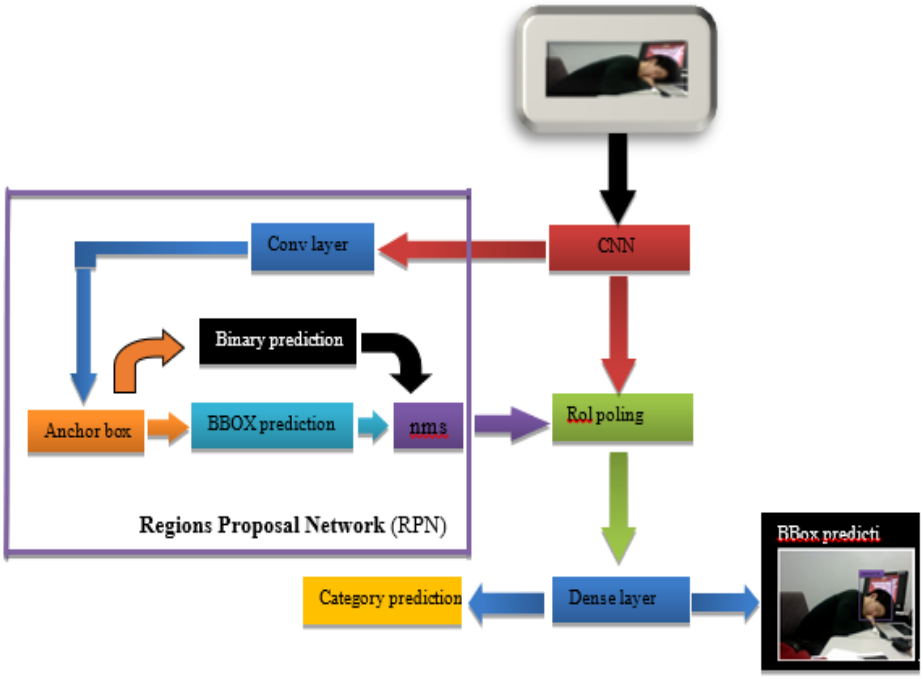

Figure 3. Faster R-CNN model with regions proposal petwork (RPN)

As we can see in Figure 3. What is he doing there? If we see input and output in figure 3.1 then hope that we would be able to understand our network. We took a picture for understanding according to that network and have described that method how they work all surrounding steps of faster R-CNN model with regions proposal network. It is worth noting that the Faster R-CNN model trains the regional suggestion network along with the rest of the model. In addition, the Faster R-CNN object function also includes the prediction of categories and bounding boxes for objects with activity detection as well as the prediction of binary categories and bounding boxes for anchor boxes in the region offer network. Finally, the region suggestion network can learn how to generate high-quality supply regions, reducing the number of proposed regions while supporting object plus activity detection accuracy. We implement this model as a convolution neural network and evaluate it using the PASCAL VOC 2012 [16] detection dataset. The network's initial convolution layers extract features from the image, while the fully connected layers predict output options and coordinates. Our specification is galvanized by the Image Net [11] model for image classification.
Our network consists of twenty-four levels of convolution, followed by two fully connected levels. Instead of the starting modules used by Image Net, we simply use 1 × 1 reduction layers, followed by 3 × 3 folding layers, just like the entire network is shown in Figure 3. We are also training a fast version of YOLO that extends the limits of fast object detection. Fast YOLO uses a neural network with fewer convolution layers (9 instead of 24) and fewer filters in these layers. Apart from the size of the network, all coaching and test parameters between YOLO and fast YOLO is the same.

2.4. Yolo and R-CNN Error Analysis

YOLO strives to locate objects correctly. The percentage of localization errors in the YOLO errors exceeds the sum of all other sources. Fast R-CNN generates much fewer localization [17] errors, but much more background errors. The highest detected 13.6% were false-positive results that contained no objects. The probability of a rapid R-CNN prediction of background detection is almost three times that of YOLO.

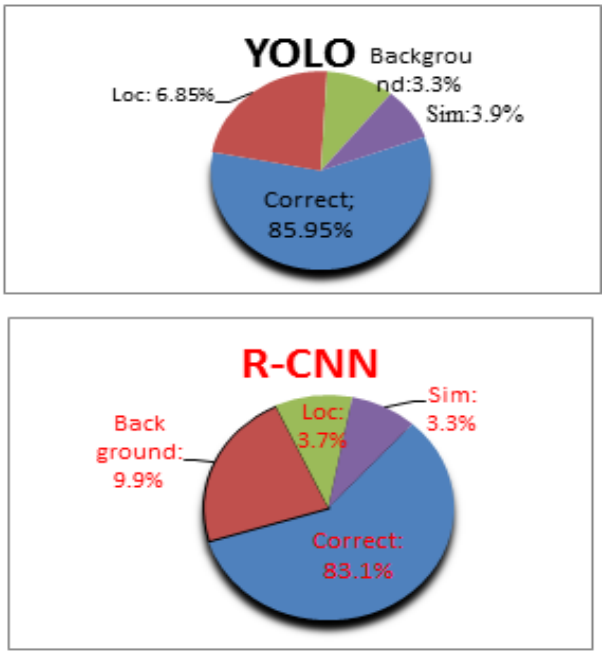

Figure 4. Error analysis- YOLO vs. R-CNN there we have shown the percentage of localization and background errors on the high N detect for several classes [N = # objects and activities of the student].

2.4.1 Combining fast R-CNN and YOLO

Compared to Fast R-CNN [18], YOLO generates far fewer background errors. By using YOLO to remove background detection for Fast R-CNN, we can significantly improve performance. For each bounding box predicted by R-CNN, we check whether YOLO predicts a similar box. In this case, we improve the prediction based on the probability of the YOLO prediction and the degree of overlap between the two fields. The mAP of the best Fast R-CNN model in the VOC 2012 test set reached 93.7%. When used with YOLO then we got the combined output of 97% shown below in Table 1. We are grouped between R-CNN and YOLO model.

Table 1. Combination of between R-CNN and YOLO

Model	mAP	Combined	Gain
Fast R-CNN [18]	90.9	92.4	-
Fast R-CNN (VGG-M) [16]	87.2	90.5	-
Fast R-CNN (CaffeNet)[8]	86.7	87.5	-
YOLOv3 [1]	93.7	95	1.2
Our R-CNN and YOLOv3	91.6	Overall 97	2.29

Model combination experiments on VOC 2012. We examine the effect of combining various models with the best version of Fast R-CNN. Other versions of Fast R-CNN provide only a small benefit while YOLO provides a significant performance boost. Our combination result on the PASCAL VOC 2012 test set, 93.3%, and YOLO received a card value of 95%. This is closer to the current state of the art the original R-CNN with YOLOv3 97 in Table 1. Our system fights with several human activities[19], compared to their closest competitors on categories like reading, phoning, using a laptop, smiling, and taking book YOLO achieves 12-15% less than R-CNN although it does not get the constant value it might be getting the exchange values. For other categories like reading, sleeping, YOLO achieves higher performance and gain 1.2 in table 5. Our combined model Fast R-CNN + YOLO is one of the best detection capability with good accuracy overall 97 shown in table 1. Fast R-CNN gets a 2.29% improvement over the combination with YOLO, boosting it 5 spots up in the SUST.

3. DATA COLLECTION AND ANALYSIS

3.1. Data Collection Scheme

There are enough target records for the detection and localization of student activities. Some of those are Stanford 40-activities, PASCAL VOC 2012 image Net [20], and Kinetics [21]. As we are focusing on detecting activity in the Library, and outdoor places of SUST, we were looking forward to using some activities class of those datasets like reading, phoning, talking book, smiling, using a laptop, etc. However, all images were from a different scenario. That is why we build a dataset called SUST_S_Act considering 12 different activities to aid the study of detecting student activities in the SUST. To build these data sets total of 10 paid assistants were associated at SUST. We have photographed them all in different locations of SUST. Our activities detection field is "Reading, Phoning, Sleeping, Taking books, and using a Laptop.

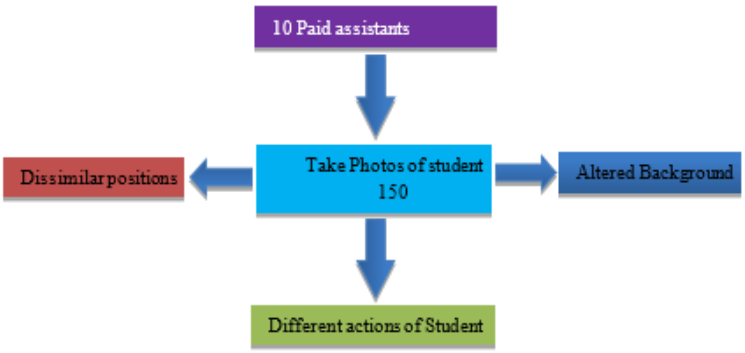

Figure 5. Data collection scheme

In the season of data acquisition, we consider several factors including the background complexity, crowded background, and the angle of view changes. We also consider multiple actions in the same image, different distances, and illumination. Thus, we collect multiple images of the different actions of a student. We made that dataset by 150 images.

Figure 6. Sample collected data of library, laboratory and field in the SUST.

3.1.1. Data Preprocessing System

First, to prepare the training data we mainly provide the ground truth activity class with a bounding box for each activity in the image [22]. For this purpose, we use the label Image, which generates a corresponding XML file for each image. The XML file includes the information about the size of the image,

its action class, the value of the bounding box (xmin, ymin, xmax, ymax). As a singular image may have several activities so there will be more than one bounding box value for those particular images.

```
<?xml version="1.0"?>
<annotation>
   <folder>sustimages</folder>
   <filename>SUST_S_Act_100.jpg</filename>
   <path>c:\users\YOUSUF\Desktop\project\student=action=model\dataset\
          train\SUST_S_Act_100.jpg</path>
   <source>
         <dataset>Unknown</dataset>
   </source>
   <size>
         <width>1000</width>
         <height>900</height>
         <depth>3</depth>
   </size>
   <degmented>0</segmented>
   <object>
         <name>studying,phoning</name>
         <pose>Unspecified</pose>
         <truncated>0</truncated>
         <difficult>0</difficult>
         <bndbox>
             <xmin>250</xmin>
             <ymin>360</xmin>
             <xmax>600</xmax>
             <ymax>750</ymax>
         </bndbox>
   </object>
</annotation>
```

Figure 7. This is first method of data preprocessing, XML file of label image.

We have used two techniques for data prepossessing. We have made these data eye-catching by using our proposed algorithm ML (machine learning) and XML file. The XML and machine learning-based algorithm those are good implicit for each image. The machine learning-based data prepossessing are big procedure method for all kind of activities and they have the best accuracy and powerful detection ability. Although this method is a little bit complex than XML as we have shown in Figure 7 and Figure 8.

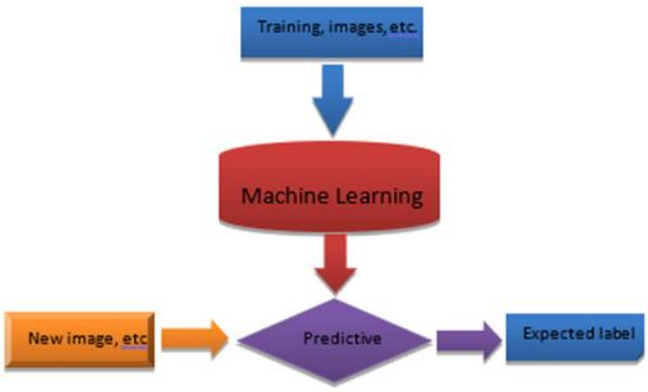

Figure 8. The second data prepossessing method

3.1.2. Activity Representation

The first and foremost important problem inactivity detection is how to represent activity in an image. Student activities appearing in images differ in their motion speed, camera view, appearance, and activity variations, etc., making activities representation a challenging problem. A successful action representation method should be efficient to compute, effective to characterize activities, and can maximize the discrepancy between activities, in order to minimize the classification error. One of the major challenges in inactivity detection is large appearance and activity variations in one activity category [23], making the recognition task difficult. The goal of activity presentation is to convert an activity image into a feature vector, extract

representative and distinctive information from student activity, and minimize the variations, thereby improving detection performance. Activity representation approaches can be roughly divided into holistic and local features, which we are discussed below. Inactivity detection many attempts have been made to convert the activity image into distinctive and representative features, minimize intra-class variations, and maximize between-class variations. At this time, we effort on some verity activities illustration methods, which the parameter is means in these systems are pre-defined by specialists. This varies from deep networks, which can habitually learn parameters since the document.

Activities prediction tasks are roughly categorized into two kinds, short-term prediction, and long-term prediction. The short-term prediction focuses on short period activity which usually last for many seconds, such as in PASCAL VOC 2012 datasets [19]. The goal of this task is to infer activity labels based mostly upon temporally incomplete images.

Formally, given associate degree incomplete

Activity image $\mathbf{X_{1:t}}$, **t** frames, i.e.$\mathbf{X_{1:t}} = \{\boldsymbol{f_1}, \boldsymbol{f_2}, \ldots, \boldsymbol{f_t}\}$,

The goal is to infer the activity label $\mathbf{Y}:\mathbf{X_{1:t}} \rightarrow \boldsymbol{y}$.

Here, the incomplete activity $\mathbf{X_{1:t}}$, contains the beginning portion of a complete activity execution $\mathbf{X_{1:T}}$, which only contains one single activity. The latter one, long-term prediction or intention prediction, infers the long run activities based on current discovered student activities. It intended for modeling activity transition, and so focuses on long-duration image that last for a few minutes. In different words, this task predicts the activity that is getting to happen in the future. More formally, given activity images $\mathbf{X_a}$wherever $\mathbf{X_a}$ could be a complete or incomplete activities execution, the goal is to infer next activity $\mathbf{X_b}$. Here, $\mathbf{X_a}$ and $\mathbf{X_b}$ are two independent, semantically meaningful, and temporally correlated activities.

3.2. Dataset analyses

This section discusses some of the popular activities video and image datasets, including activities captured in a controlled and uncontrolled environment. These datasets differ in the figure of student activities, background noise, and appearance and activities variations, camera motion, etc., and have been widely used for the comparison of various algorithms and RGB activity datasets [24].

3.2.1. Activity Datasets analyses

This is our main analysis dataset for this article. PASCAL VOC 2012 is a standard and big image dataset for detection. There are consists of 7 types of student activities such as (Reading, taking the book, phoning, sleeping and using a laptop, smiling, talking so on) repeated several times by 25 different subjects in 4 scenarios (outside, inside with scale deviation, outside with different activities and inside). There are 600 activity images in the dataset. We also have analyzed other datasets like kinetics [25], and Weizmann [26], dataset used for only popular video datasets for any recognizing of human activities. The data set contains 10 activity classes, which are carried out by 9 different topics, e.g. B. "moving", "talking", "watching" to provide a total of 3 video sequences. The video was recorded with a static camera against a static background.

The activity net [27], and UCF101 [28], data set also contain realistic videos collected from YouTube [29]. It contains 10 activities categories with a total of 20 videos. That's why we have selected the PASCAL VOC 2012 dataset on images for student activities detection. PASCAL VOC 2012 dataset has the greatest variety of motion, and there are big differences in terms of camera movement, object appearance, and posture, object ratio, viewing angle, overloaded background, lighting conditions, etc.

Table 2. A list of popular activity image and video datasets used in activity detection research.

Datasets	Years	Actions	Modality	Env.
PASCAL VOC 2012[10]	2019	25	RGB-D	Controlled
WEIZMANN [26]	2005	10	RGB	Controlled
KTH[27]	2004	6	RGB	Controlled
KINETICS[25]	2017	600	RGB-D	Uncontrolled
ACTIVITY NET [24]	2015	203	RGB	Uncontrolled
UCF 101 [28]	2009	1,100+	RGB	Uncontrolled
CA [17]	2009	44	RGB	Uncontrolled
MSR-I [18]	2009	63	RGB	Controlled
MSR-II [19]	2016	54	RGB	Crowded
MHAV [20]	2017	238	RGB	Controlled
UT-I [29]	2018	60	RGB	Uncontrolled
TV-I [32]	2019	300	RGB	Uncontrolled

4. EXPERIMENTAL RESULTS AND DISCUSSION

We do the experiments based on our test data to find out whereby enormously we can detect accurately. To do our experiment, we trained our model at first so that it can recognize notable features of our actions from the training dataset. We need to do fine-tuning to our model to get better accuracy regarding detection purposes. We get the overall efficiency of 97% after training of the 150th iteration. This efficiency further depends on the amount of iteration we perform. In the next section, we will show the accuracy of the training set, the validation set, and the test set. For the experimental results, we need to calculate the accuracy of the task of detecting student actions in the library. For this, we used the method of Average Precision and Mean Average Precision. Average Precision is used to calculate the accuracy of actions independently. While means Average precision is used to calculate the accuracy of activities in a combination. Most of the time we need to work with different angles of the same activities. For this, feature extraction is quite difficult for the detector. We need to consider all the possible features. With the digitalization of the computer, there have been many efficient techniques to perform the detection task. We trained our model with the dataset of "SUST_S_Act" to detect five activities. The first activity is "Reading" got the three precision is 98.2% table 3, 98.05% table 4, and 98.3% table 5, the second activity precision is "taking the book" 96.5% table 3, 93.6% table 4, and 94.6% table 5. The third activity precision is "Phoning" 97.09% table 3, 97.1% table4, and 97.7%, table 5. Fourth activity is "Sleeping", 97.1% table 3, 97.9% table 4, and 97.5% table 5. And the last activity is "using a Laptop" 95.99% table 3, 96.5% table 4, and 96.97% table 5. In our model, there are several stages to extract features from the "SUST_S_Act" image dataset and runs many times to get a better result. In the meantime, training, fine-tuning is beginning. When our model parameters will get the "fine-tuning", we will able to detect activities more accurately. When we completed the 3rd iteration, our model stopped training because there is no updating in the last three iterations. We got an overall 0.97 mAP which is called 97% activity precision. We find out mAP for train, validation, and test data. All of them are near and overall at 0.97. The table of mAP values for the train table 3, validation table 4, and test data table 5 are given below.

Table 3. Training set accuracy (IoU 0.5)

STUDENT ACTIVITIES	ACTIVITIES PRECISION	ACTIVITES AVERAGEPRECISION
Reading	98.2%	
Taking the Book	96.5%	
Phoning	97.09%	
Sleeping	97.1%	0.9707 = 97%
Using a Laptop	95.99%	

Table 4. Valid set accuracy (IoU 0.5)

STUDENT ACTIVITIES	ACTIVITIES PRECISION	ACTIVITES AVERAGEPRECISION
Reading	98.05%	
Taking the Book	93.6%	
Phoning	97.1%	
Sleeping	97.9%	0.9743 = 97%
Using a Laptop	96.5%	

Table 5. Test set accuracy (IoU0.5)

STUDENT ACTIVITIES	ACTIVITIES PRECISION	ACTIVITES AVERAGEPRECISION
Reading	98.3%	
Taking the Book	94.6%	
Phoning	97.07%	
Sleeping	97.5%	0.9701 = 97%
Using a Laptop	96.97%	

4.1. Descriptive Analysis

We did divide the dataset into three parts. They are the train (Table 3), validation (Table 4), and test data (Table 5). Train data contains 97% of the "SUST-S_Act" whereas validation and test data contain was 95% after that we would try our best to get the same accuracy of the "SUST-S_Act" each dataset. Finally, we

are able to increase by 2% for an equal dataset. We evaluate our model by test data that is unique from the train and validation set. Our model is already introduced, and now our model overall accuracy of 97% (table 3, 4, 5). And also analyzed IoU performance of true and false results show in Figure 9.

$$\mathbf{IoU} = \frac{\textbf{Area of Intersection}}{\textbf{Area of Union}} \quad (6)$$

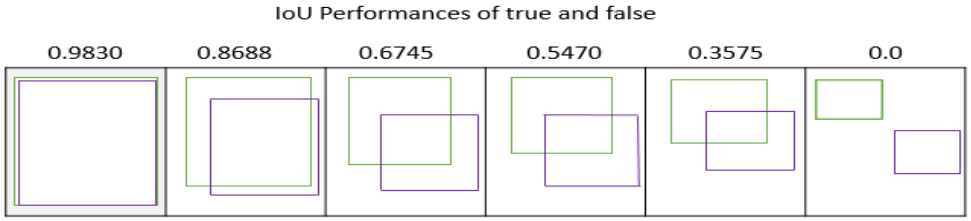

Figure 9. IoU performances of true false

As we can see in Figure 9 the highest true precision of 0.9830. It indicates that our detector can detect the reading activity with good accuracy of 98.3% table 3 and Fig 10. Even though we got there some lass true detection performances of IoU such as 0.6745, 0.5470 and also absolutely false detection result is 0.3575. Therefore, we took her the best results from them and also discussed them. Let us see a detection output of "Reading".

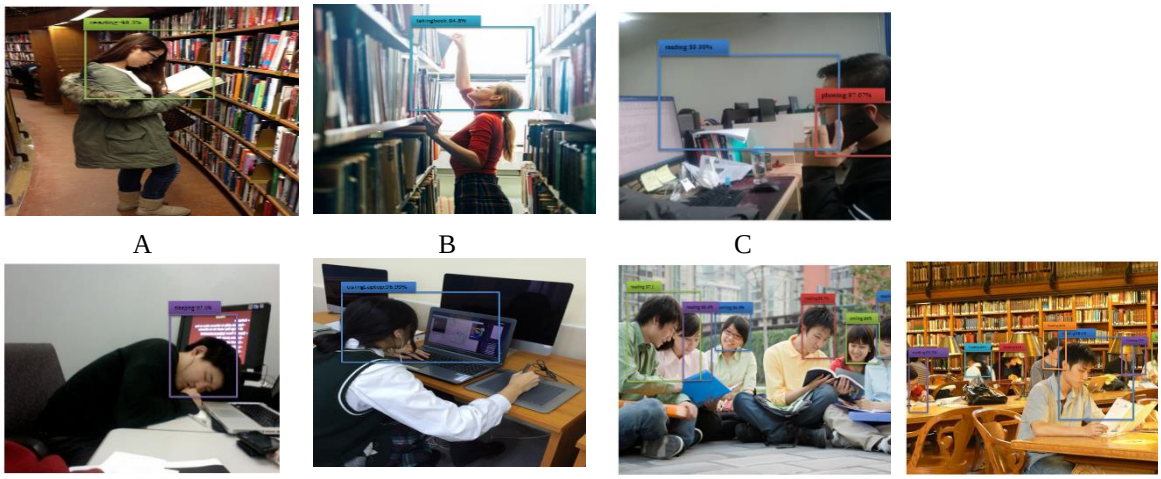

Figure 10. Our experimental result of activity detection in single and multi-image reading, taking book, phoning, sleeping and using laptop

And in table 5, as we can see here that taking the book has the lowest average precision of 96.5 % table 3, 93.6% table 4, 94.6% table 5. It indicates that our detector can detect the "taking the book" activity with an accuracy of 94.6%. Let us see a detection output of "taking the book". We also see that phoning, sleeping, and using a laptop has an accuracy of 97.07% table 3, 97.5 % table 4, and 96.99% table 5 respectively. It means our model's overall accuracy is 97%. Let us see some of the detection output of "phoning", "sleeping" and "using a laptop". Although in Figure 10-G, the person is phoning precision is 97.07% while reading precision is so lowest 89.99% Figure 10-F because here some detection problem. Our method did not detect exactly what he is doing here browsing pc or reading the book really it was a confusing thing. As we got high accuracy in every class. Our trained model can detect all-action even in an overfilled image where people are involved in multiple activities. The image in Figure10-G shows that multiple people's activities are detected so accurately even when an activity from a different class exists in the image such as reading high 98.7% smiling low 95% Fig 10-F and 10-G. We could detect in this image multi activities at a time and some activities like reading, but our method detected only 3 activities properly. Such as reading 98.1%, looking 95.5% Fig 10-G, another student activity detection is less accurate because there are background activities a little bit noisy and unclear. Though it is tough to detect activities from still images, we got satisfactory results for the detection of student activities. We got good accuracy of IoU 0.9830 to detect the "Reading" activity. We got the accuracy to detect of "using a Laptop 93.6% table 4 and taking book" 95.99% table 3 activities. However, our overall mAP is 97%, which is a good number. We did the detection of student activities such as reading, talking,

looking, phoning, sleeping, using a laptop etc. We cannot differentiate the reading and writing. Cause both are many similar activities. I hope we will overthrow it in the near future.

4.2. Evaluation Protocols for activity detection and prediction

Due to dissimilar application purposes, activity detection and prediction methods are evaluated in dissimilar ways. Shallow action recognition methods such as [24], [25], were usually evaluated on small-scale datasets, for example, Weizmann dataset [26], KTH dataset [27]]. Leave-one-out training scheme is popularly used on these datasets, and confusion matrix is usually adopted to show the recognition accuracy of each action category. For sequential approaches such as per-frame recognition accuracy is often used. UCF-101 [28] and Deep networks [29], are generally evaluated on large-scale datasets thus can only report overall recognition performance on each dataset. In [30], average precision that approximates the area under the precision-recall curve is also adopted for each individual action class.

There are numerous popular metrics for estimating activity detection prediction methods, including Average Displacement Error (ADE), Final Displacement Error (FDE), and Average Non-linear Displacement Error (ANDE). ADE is the mean square error computed over all estimated activity of student and the specific object-truth point out. FDE is defined as the distance between the predicted final destination and the object-true final destination. ANDE is the MSE at the non-linear turning regions of student activities arising from student -object interactions.

4.3. Implication of Further Studies

Each system has been forming with upcoming progressing opinion. In the future, our system will be faster and more efficient. Reducing processing time is one of the important issues. We will be developed for better performance since now. We want to continue the research in this field. We will try to detect more complicated actions. We will work with videos. We will try to get more accuracy by applying various techniques. We will resolve to make a assimilate software by which we may mark a report of people's activities. The automatic alarming and sensor-based system will be unindustrialized for uninvited activity. Each system has been forming with upcoming progression prospect's opinions. In the future, our system will be faster and more efficient. Reducing processing time is one of the important issues.

5. CONCLUSION

The accessibility of big data and significant models diverts the research effort about student activities from understanding the current to reasoning the future. We have presented a complete article of state-of-the-art techniques for activity detection and prediction from images. These techniques became particularly interesting in recent decades due to their promising and real-world presentations in several emerging fields focusing on student activities. We investigate some aspects of the prevailing attempts including YOLOv3 and R-CCN feature design, models and algorithms, datasets, and system performance. Future research directions are also discussed in this paper.

REFERENCES

[1] N.https://awesomeopensource.com/projects/yolov3.
[2] G.Gkioxari, R. Girshick, and J. Malik, "Contextual action recognition with R*CNN," in Proc. IEEE Int'l Conf. on Computer Vision, 2015.
[3] https://www.researchgate.net/publication/326535574_YOLO_based_Human_Action_Recognition_and_Localizatin W. Choi, K. Shahid, and S. Savarese, "Learning context for collective activity recognition," in CVPR, 2011.
[4] Y. Cao, D. Barrett, A. Barbu, S. Narayanaswamy, H. Yu, A. Michaux, Y. Lin, S. Dickinson, J. Siskind, and S. Wang, "Recognizing human activities from partially observed videos," in CVPR, 2013.
[5] B. G. Fabian CabaHeilbron, Victor Escorcia and J. C. Niebles, "Activ-itynet: A large-scale video benchmark for human activity understand-ing," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 961–970.
[6] M. Jain, J. Van Gemert, H. Jegou, P. Bouthemy, and C. G. "Snoek. Action localization with tubelets from motion" In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 740–747, 2014.
[7] B. Yao and L. Fei-Fei, "Recognizing human-object interactions in still images by modeling the mutual context of objects and human poses," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 34, no. 9, pp. 1691–1703, 2012.
[8] Duarte, Kevin & S Rawat, Yogesh & Shah, Mubarak. (2018). "VideoCapsuleNet: A Simplified Network for Action Detection.

[9] https://www.oreilly.com/library/view/python-deep-learning/9781789348460/69c62e2a-c8a4-480d-a364-91ecd4d7199b.xhtml.
[10] G.Rizzolatti and C. Sinigaglia, "The functional role of the parieto-frontal mirror circuit: interpretations and misinterpretations," Nat. Rev. Neurosci., vol. 11, pp. 264–274, 2010.
[11] H. Wang, D. Oneata, J. Verbeek, and C. Schmid, "A robust and efficient video representation for action recognition," IJCV, 2015.
[12] T. Lan, L. Sigal, and G. Mori, "Social roles in hierarchical models for human activity," in CVPR, 2012.
[13] J. Yuan, Z. Liu, and Y. Wu, "Discriminative subvolume search for efficient action detection," in IEEE Conference on Computer Vision and Pattern Recognition, 2009.
[14] J. L. Jingen Liu and M. Shah, "Recognizing realistic actions from videos"in the wild"," in CVPR, 2009.
[15] K. K. Reddy and M. Shah, "Recognizing 50 human action categories of web videos," Machine Vision and Applications Journal, 2012.
[16] https://towardsdatascience.com/coco-data-format-for-object-detection-a4c5eaf518c5 Hou, Rui & Chen, Chen & Shah, Mubarak. (2017).
[17] M.Bregonzio, S. Gong, and T. Xiang, "Recognizing action as clouds of space-time interest points," in CVPR, 2009.
[18] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt. "Sequential deep learning for human action recognition. In Proceedings of the Second International Conference on Human Behavior Unterstanding" HBU'11, pages 29–39, 2011.
[19] Simonyan, Karen & Zisserman, Andrew. "Stream Convolutional Networks for Action Recognition in Videos" Advances in Neural Information Processing.
[20] C. Wang, Y. Wang, and A. L. Yuille. "An approach to posebased action recognition" In Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, CVPR '13, pages 915–922, Washington, DC, USA, 2013. IEEE Computer Society.
[21] K. Soomro, H. Idrees, and M. Shah. "Action localization in videos through context walk" In IEEE International Conference on Computer Vision (CVPR), pages 3280–3288, 2015.
[22] V.Delaitre, J. Sivic, and I. Laptev, "Learning person-object interactions for action recognition in still images," in Proc. Advances in Neural Information Processing Systems, 2011.
[23] J. Sung, C. Ponce, B. Selman, and A. Saxena, "Human activity detection from rgbd images," in AAAI workshop on Pattern, Activity and Intent Recognition, 2011.
[24] Schmid, Cordelia. "AVA: A video dataset of spatio-temporally localized atomic visual actions" CVPR, 2018.
[25] Eren, M. Baccouche, E. Dellandrea,´ C.-E. Bichot et al., "Evaluation of video activity localizations integrating quality and quantity measure-ments," Computer Vision and Image Understanding, vol. 127, pp. 14– 30, 2014.
[26] C. Schuldt,¨ I. Laptev, and B. Caputo, "Recognizing human actions: A local svm approach," in IEEE ICPR, 2004.
[27] M. D. Rodriguez, J. Ahmed, and M. Shah, "Action mach: A spatio-temporal maximum average correlation height filter for action recogni-tion," in CVPR, 2008.
[28] A. R. Z. Khurram Soomro and M. Shah, "Ucf101: A dataset of 101 human action classes from videos in the wild," 2012, cRCV-TR-12-01.
[29] M. Marszałek, I. Laptev, and C. Schmid, "Actions in context," in IEEE Conference on Computer Vision & Pattern Recognition, 2009.
[30] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," in Proc. ICCV, 2005