Polynomial Regression and Faster R-CNN Models for University Library Decision Implementation Discovery based on Deep Learning

SHKELQIM HAJRULLA[1], ÖZEN ÖZER[2], TAYLAN DEMIR[3]
[1]Computer Engineering Department, Faculty of Engineering,
Epoka University, Tirana,
ALBANIA

[2]Department of Mathematics, Faculty of Science and Arts,
Kirklareli University,
TURKEY

[3]Department of Mathematics, Faculty of Sciences,
Ankara University,
TURKEY

Abstract: - We deal with the development of a seat occupancy detection algorithm for the University's library utilizing the Faster R-CNN algorithm. The university's library is widely used by students, particularly during exam season when it can become difficult to find a seat. Going from one building to another is often time-consuming and useless when there are no available seats. This system uses the features of Faster R-CNN in a way to facilitate an automatic seat occupancy monitoring system. Unlike conventional methods of using manual monitoring or weights and occupancy switches as inanimate indicators, it provides real-time seat availability data which allows no human intervention to be a part of the process. Data collection and model training evaluation are considered using annotated datasets with images of library seating layouts. The Faster R-CNN model is trained such that it can accurately detect vacancy or occupancy at library seats. This work takes a futuristic approach towards smart library management systems, in which user needs are changing, and considers the use of high-end computer vision technologies to be integrated into all such libraries. The proposed system aims to leverage the effectiveness of Faster R-CNN and go a long way in redefining seat occupancy management for university libraries by enabling better efficiency, resource utilization, as well as user satisfaction in prospect.

Key-Words: - Numerical Models, Object Detection, Math Regression; Convolutional, R-CNN Model, Model Training and Evaluation, Computer Vision, Annotated Datasets, Efficiency in Seat Management, Futuristic Library Systems.

Received: April 12, 2024. Revised: November 9, 2024. Accepted: December 11, 2024. Published: January 7, 2025.

1 Introduction

With the growing emphasis on resource usage in sites such as libraries, never has been the time to provide quality education with less weight. We call the technique of watching and re-situating seats within our library (or classroom) "seat occupancy management." We aim to study the implementation ability of successful seat occupancy coordination via models and process evaluation. Ensure that spaces are fully utilized and that students can be safely and comfortably accommodated. It is also a key piece when it comes to reducing overhead and enhancing the overall experience. This is particularly important when it comes to whether users are occupying a seat or not, such as during peak travel periods like exam seasons. In the wake of this problem, an automated solution involving modern-day technology to perfect space utilization has become prevalent as a method for administration and monitoring stock in real time.

Our research focuses on the development of a seat occupancy detection algorithm tailored for the University's library. By harnessing the capabilities of the R-CNN algorithm and analyzing convolutional process [1], the proposed system offers a sophisticated approach to real-time seat occupancy monitoring, eliminating the need for constant human intervention. The research holds great importance as it has the ability to transform the administration of library space, optimize

resource allocation, and improve the user experience in general. This article attempts to enhance the continuous development of smart library management systems and convolution processes applying several filters, also referred to as kernels by integrating cutting-edge computer vision technologies.

The authors [2] propose Action Progression Networks (APNs) for temporal action detection in videos. APNs address the challenge of detecting actions with varied temporal lengths by modeling action progression through time. The networks improve the detection accuracy of complex video datasets by incorporating progression networks and temporal dependencies. This approach surpasses existing benchmarks and enhances the performance of video action detection. It is particularly useful for applications like video surveillance and sports analysis.

Any successful computer vision project starts with a high-quality, robust dataset that is utilized for both training and assessment. When it comes to creating precise and effective algorithms for seat occupancy detection using a learning system [3], in academic settings, having access to a well-annotated dataset is essential.

The authors in [4] introduce a new speech coding method using CNNs to enhance speech recognition systems. By encoding speech signals through convolutional neural networks, the method improves the accuracy of speech recognition, especially in real-time systems. The approach maintains a balance between signal symmetry and processing efficiency, making it suitable for applications in digital assistants and telecommunication systems. This research demonstrates the potential of CNNs in fields beyond visual tasks, extending their applicability to audio processing.

The work [5] focuses on temporal action detection in videos, employing a statistical language model to improve action localization. By predicting and modeling action durations within video sequences, the system enhances the precision of temporal detection. The combination of language models with action detection offers significant improvements over conventional methods. This technique is highly beneficial for applications that require precise timing of action occurrences, such as video indexing and content moderation.

Current systems for speech recognition are based on deep learning methodologies [6], which employ both feature representation and language models as the main targets of this work. With such a challenge, this article dives into some of the most prospective ways of computer vision. Faster R-CNN model among many others, some of which are trained nimbly and the speech identification system is the main method based on variable real - time monitoring. In real-time detection, the fastest efficiency of the model is investigated in this context.

This article explores the dataset preparation process that was carried out in order to produce an extensive dataset that was specially designed for the library at the university and the convolutional neural networks process is very important and implemented during this research. Our method processes [2] untrimmed videos on a deduction seat basis to fit the R-CNN method.

The Microsoft COCO (Common Objects in Context) dataset is introduced in the paper [7], serving as a large-scale benchmark for object detection, segmentation, and captioning. COCO is designed to reflect real-world image complexity by including context and object interactions. The dataset has become a standard in the field, enabling researchers to develop more robust and accurate models. The paper underscores the importance of context in object recognition and drives innovation in computer vision.

The study [8] introduces the Temporal Unit Regression Network (TURN-TAP) for generating temporal action proposals in videos. The model improves the accuracy of detecting temporal boundaries in video actions by leveraging regression techniques. TURN-TAP outperforms previous methods in terms of both speed and precision, making it ideal for real-time video analysis. The approach is especially useful in fields like video summarization and action recognition in dynamic environments.

The authors in the work [9] propose a deeply supervised salient object detection model with short connections, which improves object detection performance by using hierarchical feature maps. The model enhances the detection of salient objects by incorporating short connections to maintain spatial consistency across layers. This method addresses challenges in accurately detecting visually complex objects, making it suitable for applications like medical imaging and autonomous systems. The paper contributes to improving the overall efficiency of salient object detection models.

The paper [10] presents a hybrid fully convolutional network (FCN) for actionness estimation, aiming to estimate the likelihood of actions occurring in video frames. By combining spatial and temporal features, the model enhances the accuracy of detecting action proposals. The

approach demonstrates high precision in real-world video datasets, providing insights into both action classification and temporal segmentation. This work is influential in the development of video content analysis tools.

The authors in [11] rethink the Faster R-CNN architecture for temporal action localization, proposing improvements to adapt the architecture for video analysis. They enhance the temporal detection accuracy by incorporating long-term temporal features into the model. This method significantly improves the detection of action boundaries in untrimmed video sequences. The paper sets a new benchmark in video action localization and is instrumental in advancing video content understanding technologies.

The need to design a Seat Occupancy Detection Algorithm for the university library is due to realizing the challenges of managing seating in a busy academic environment. In the example given, this particular problem of keeping a check of available seats becomes even more critical when academics and students are looking for joint study and workspaces. But, the problems of effectively detecting occupied seats, recognizing goods left behind on tables and chairs as well and figuring out the presence of human beings are very complicated some of which are trained nimbly and speech identification systems. Efficient evaluation of convolution and detection-related problems [12], detectors, and positions of seats are analyzed in this research.

The dataset is created by collecting video footage of several scenarios that are expected to occur in the library setting. We use these videos to extract frames and do not believe that the diversity of situations captured can be fully represented by a hand-selected gallery. It also emphasizes the potential influence of this study using algorithms, images, and models on such high-performing R-CNN models as other models. Through image annotation, every frame is labeled to distinguish between seat regions, objects on tables and chairs, and human presence, providing the foundation for further algorithm development and optimization [6]. An important tool used for facilitating the labeling process and the organization of the dataset is Rob flow. Leveraging the COCO JSON format [7], the annotated dataset is structured in a standardized manner, ensuring compatibility with deep learning frameworks and ease of integration into the algorithm pipeline.

In our work, we investigate and leverage language modeling for capturing the contextual structure deduction [5] to model temporal action deductions. As we begin the process of preparing the dataset, it becomes evident that the creation of a high-quality dataset serves as the foundation of our effort [9] to develop a Seat Occupancy Detection Algorithm tailored for the University's Library.

2 Problem Formulation

The process of collecting video data for the seat occupancy detection algorithm involved a combination of careful planning, collaboration with friends, and leveraging available resources to simulate processes within the library setting. In a comprehensive literature review, the object detection process is presented, specifically in the case of library surveillance. We established three main goals: assessing the efficacy of Faster R-CNN in the detection of objects [6] contrasting their capabilities with respect to accuracy, speed of execution based on remote sensing imagery with deep learning methods, and the ability to adapt to changing environmental conditions. To address this problem, we built our model using the Temporal Unit Regression Network Model (model is also mentioned in [8]) for the purpose of our regression processing our research we split data up in the format 60/20/20 for training/validation/testing. Regression class is used to train a regression model.

2.1 Data Collection

Due to the unavailability of access to library cameras, a pragmatic approach was adopted, where 24 videos were captured using a smartphone camera. These videos were carefully crafted to encompass a range of scenarios commonly encountered within the library environment, ensuring the diversity and representativeness of the dataset building the R-CNN model. It understands deep neural networks [9] which are utilized for object classification. In particular, because of its popularity in obtaining detections of images, we aim to investigate if the R-CNN model is suitable for the localization of space. Collaboration played a pivotal role in the dataset collection process, with some students simulating various seating arrangements and human interactions. We use prior information and then apply the action detection [10] proposed in which many bounding boxes is the estimated maps for seated position. Each video scene featured a table with four seats on each side, mirroring the typical layout found in the library's study areas (Figure 1). The scenarios enacted by my friends were carefully orchestrated to capture a spectrum of situations, including:

1. All students are seated to study, simulating a fully occupied seating arrangement.
2. While some students are seated; partial occupancy of seats
3. Students get up from their seats but leaving all their stuff right at the table or on their chairs, pretending as though only they would be taking part in a transitory occupation.
4. For example, one covered the behaviors of students engaging in interactions (approaching another seated individual but not taking a seat themselves).

In this way the dataset collection process was designed to elicit a depth of complexity more reflective of authentic library interactions, as students attempted real—it—knew it! Speech recognition approach using Convolutional Neural Network: Speech coding for speech recognition with method. The choice to use video data as the main kind of dataset took into account several reasons. For starters, videos provide a holistic and dynamic view of real-world situations with changes over time along with the interactions between humans, which static images may lack. This temporal dimension is also critical in training the seat occupancy detection algorithm to differentiate transient in a case where some people leave for a moment but are keeping their belongings on the seat (and hence, not available for others).

Also, the videos selected for the dataset were chosen with the idea that real-use libraries will make use of surveillance type cameras already within the library culture to implement their seat occupancy detection. While the surveillance camera footage data would have given a wider shot, we decided to use cell videos for data collection as they offer better access.

2.2 Extraction of Frame

During frame extraction each frame from each in the dataset was systematically collected to assemble a visual data pool. With the help of Open CV, they had to iterate over the frames of videos and save each frame as an independent image file to an output folder. Both solutions provide organization during naming as well as a logical sequence for referencing tasks. Sequential naming ensured organization and ease of reference. Throughout the extraction process, the existence of the output folder was verified, and resources were released after each video to optimize memory usage. This procedure resulted in the extraction of 4253 frames, each representing a snapshot of different scenarios within the library environment, including seating arrangements, human interactions, and objects.

2.3 Splitting of research dataset

The model is prevented from overfitting; that is, learning the data by heart. Data must be shared into 3 parts: training, validation, and testing data.
Providing photos to a computer vision model allows it to be taught. The model uses a function to estimate how close it is to making the correct prediction. The model builds a prediction function to the underlying loss function that causes the solution to assign an output to every pixel in the image. It is possible to be adapted during training. For example, the model may learn a very reducible function that works very fast during the training time process (Figure 2, Figure 3 and Figure 4).

The loss function on the training data will keep showing lower and lower values if the model hyper-specifies to the training set, but it will finally increase on the held-out validation set. We used Roboflow to partition the dataset close to 70% of the data for the process of training, 15% validation, and 15% test of the results. As a result, the training set is shown in Figure 1. Train 2980 images, validation 640 images, testing 632 images.

Fig. 1: Dataset insights before annotation

The model is trained on progressively larger subsets of the training data and for each subset. Also the computation of Mean Squared Error (MSE) for the training & validation set. We perform fitting a polynomial regression to the data we've generated and visualize the results. We generate synthetic data from a known quadratic function, cubic function, and the function of degree 6, See the following figures (Figure 2, Figure 3 and Figure 4).

It means that this model will be a bad result in the case of a novelty. It has never been seen before since it is not learning well and is essentially memorizing the training set [Train, validate, test split for machine learning, lastly viewed on 2 June 2024].

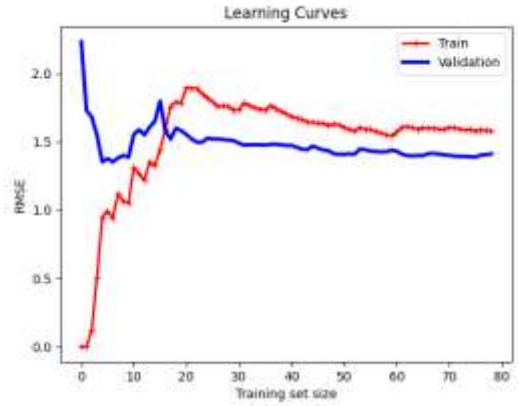

Fig. 2: Fitting a quadratic polynomial regression to the data [Train, validation, test split for machine learning, lastly visited on 2 June 2024]

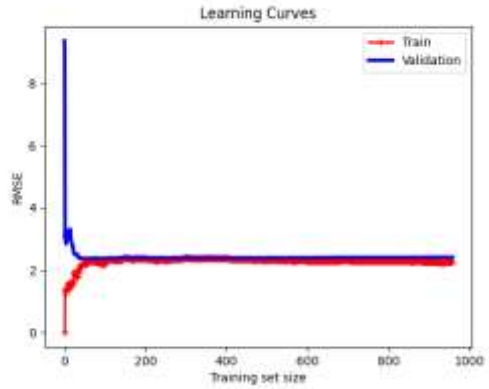

Fig. 3: Fitting a cubic polynomial regression to the data [Train, validation, test split for machine learning, lastly visited on 2 June 2024]

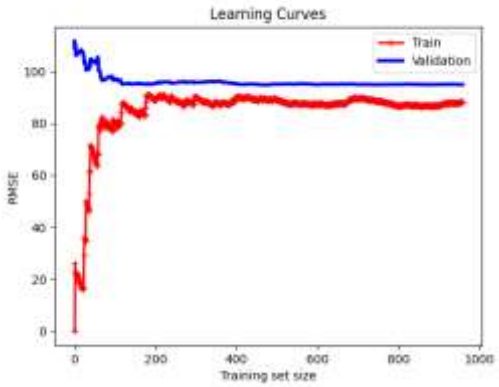

Fig. 4: Fitting a polynomial regression of degree 6 to the data [Train, validation, test split for machine learning, lastly visited on 2 June 2024]

2.4 Dataset Annotation

We continue with the preparation of the dataset using Roboflow. In the beginning, we tried to develop a model to automatically annotate the images since there is a considerable amount of images to annotate. We manually annotated approximately 200 images and fed them to the network to learn from. Although it sounded promising in the beginning with an accuracy of more than 80%, it still didn't give very accurate results.

Therefore we decided to manually annotate all the images, which is also the main reason why we didn't create a larger dataset. Although manually annotating images is a time-consuming process, the results being very precise highly benefit the model.

The dataset is annotated using three classes: seat region, human, and object. Seat regions are defined as the area encompassing a chair and the table space in front of that chair. Human annotations aim to identify individuals within the image frame. Object annotations focus on identifying items left on tables and chairs. This encompasses various objects commonly found in study environments, such as laptops, bags, and books. For each seat region, human, and object of interest, we have drawn bounding boxes. The annotation files are formatted according to the COCO JSON format, a widely used standard for image annotation in computer vision tasks.

In this format, each annotation file contains structured data representing the annotated images, including information about the image file, annotations for objects of interest, and metadata such as image dimensions. *A* summary of the dataset after being annotated can be seen (Figure 5), 4253 images are annotated in total, with an average of 15.5 annotations per image and 65726 annotations overall. The median image ratio is 480x848, which is also the size for each image.

Fig. 5: Dataset insights after annotation

3 Deep Learning Approaches

The convolution's operation is depicted in the graphic below. The patch measures 3 by 3. The output matrix is the outcome of the element-by-element manipulation performed by the filter on the picture matrix.

3.1 Conceptual Model

ConvNets are a subclass of ANNs named artificial networks that are characterized by their deep feed-forward architecture and faster generalization than networks that only have fully connected (FC) layers. It's built to learn spatial feature hierarchies

automatically and adaptively, from basic patterns to complex ones. The CNN model ((model is also mentioned in [4]) consists of a limited number of processing layers, where every layer allows the model to learn different features from input data (here image) at different levels of abstraction.

Deep layers are used to learn more about levels whereas abstraction is used from the use of initial conditions to keep the knowledge level and to focus on the improvement of high-level images. Figure 6 depicts the basic conceptual model of CNN.

3.2 Convolutional

The convolution layer is a very important process of the convolutional. The majority of the computational load is placed on it. Convolution is the process of applying several filters in the context of CNNs. These filters convolve tiny matrices of learnable parameters with the input using a sliding window technique. The filters are designed to look for specific traits or motifs in the input. For the convolution process, our research is based on some input matrixes, and kernels, and as a result, we get the output matrix of applying the convolution operation.

The convolution action can be explained as follows: the necessary filter elements and the input are multiplied element-wise and added together for each filter window location (Figure 6).

This produces a single value that is indicative of the output at that location. Swiping the filter across the whole input creates a new feature map. The image is decreased in size after convolution.

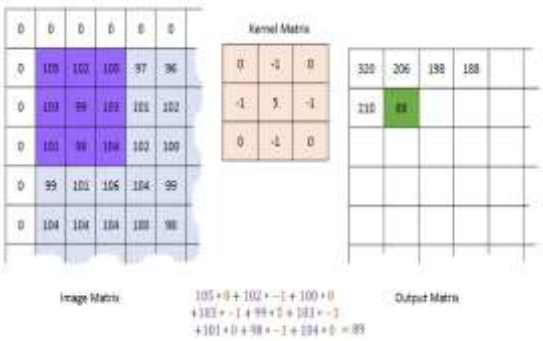

Fig. 6: Convolution action and the output matrix

Let’s explain better.

```
input_matrix = np.array([[105, 102, 100],
                         [103, 99, 103],
                         [101, 98, 104]])
```

```
Output Matrix:
[[89]]

[Execution complete with exit code 0]
```

A little array of pixels in the image will receive the filter's application during the convolutional phase. The filter will follow the supplied image in a 3x3 or 5x5 overall shape. This indicates that the network will perform the convolution by sliding these windows across the entire input image.

The process of choosing a filter (Pooling function) is applied to the feature maps. This is essentially part of a pooling layer that operates on each of those feature maps individually and creates an equivalent number of image maps in parallel. The image map is bigger than the process filter (often a 2×2 pixel filter with a 2-pixel stride) so that (a square subset of pixels in the constrained area around the portal) together with acquire, it will reduce each feature map.

Pooling always contributes to the representation being roughly invariant to tiny translations of the input. But even though this pooling strategy frequently yields positive results, the pooling layer's primary disadvantage is that it occasionally degrades CNN's overall performance.

3.3 The Fully Connected Layer (FC) and Faster R-CNN for Object Detection

Neurons from one layer to another are connected by a completely connected layer, which is made up of neurons, weights, and biases. When using training to classify photos into multiple categories, this layer is essential. It is usually the last layer in a CNN and is sometimes referred to as the output layer since it generates the final classification results. This layer would be used for creating a vector n-dimensional output, using it during the program selects.

Recent developments in Faster R-CNN-based models (model is also mentioned in [11]) have made them a significant area of research due to the increasing demand for accurate and dependable object recognition. In order to get around the repeated computation of convolutional features, a revised network called Faster R-CNN is presented.

Three primary components are used in this two-stage architecture to provide end-to-end training. To extract generalized features, it uses a unified convolutional backbone; to make class-agnostic proposals, it uses an efficient Region Proposal Network (RPN); and to construct task-specific RPNs.

4 Application

The paper builds on the Faster R-CNN framework, by simply deploying a popular Fast R-CNN detector with a Region Proposal Network, supported by a

backbone ResNet-50 and Pyramid Network for strong feature representation and accurate object detection. The first step is to import all the important libraries.

4.1 Processing the Data for the Research Model

Load the model and process the data in COCO dataset format, bounding boxes are in the form of coordinates as width-x, height-y. On the other hand, Faster R-CNN needs bounding boxes in [x_1, y_1, x_2, y_2] format specifying top-left and bottom-right corner coordinates. An object to be found by web-based image investigation [10], is almost always represented in a side aspect, and thumb impressions are our regression together with calculations when we perform an investigation.

The class Custom Dataset performs better for the training process, validation, and testing related to the desired images and annotations. This step will format every set in a way to use it for further processing and for model training & evaluation. The collate function takes a batch of data which is tuples of images, bounding boxes, and labels as input. Then it combines all the images into a single tensor and provides them as well as bounding boxes and labels one by one.

This is important to bear in mind when we construct a custom collate function, with the goal of preserving the appropriate structure and format of newly batched data. Lastly, Data Loaders for the training and validation datasets are created using these transforms. The loaders are initialized with the datasets, batch sizes, and other info such as the custom collate function.

The batch size 4 is placed for a train loader and is set to 1 for validation and test loaders. The train loader will shuffle the data, which is needed to break any potential order in the training and guarantee more robust learning. However, note that validation and test load are not shuffled because data is provided in order (for testing the seq2seq, we must have all outputs for a given input at inference time).

4.2 Optimizer

To minimize the loss function, we proceed by updating the model weight function, creating the possibility of comparison results, and measuring how well the model's predictions agree with actual goals.

The optimizer aids in the model's prediction improvement by reducing the loss. It states that the learning rate is 0.005. The limited number of steps allows the optimizer to optimize to the minimum of the loss function determined by the learning rate.

A small learning rate makes the training process slow, but it may lead to more precise convergence. The momentum is specified as 0.9. Momentum helps accelerate the optimizer in the right direction by considering the previous gradients. It helps smooth out the updates and may prevent the optimizer from becoming trapped in local minima. The specified weight decay is 0.0005.

A regularization technique called weight decay adds a term to the loss function that promotes smaller weights, hence assisting in preventing overfitting. Additionally, a learning rate scheduler is made in the optimizer function. A Stapler Scheduler is the kind of learning rate scheduler. With each step size epoch, this scheduler lowers the rate where the step size is three, and the gamma is given as 0.1.

The classification loss, which quantifies the error in class labels for the objects, and the regression loss, which quantifies the error in bounding box coordinate prediction, are two of the components that make up the loss function. The defined function in the algorithm is responsible for summing each of the individual loss values that the loss dictionary returns in order to calculate the overall loss.

Understanding the research epoch where their number of training models is chosen to be 20, meaning that the training process is repeated 20 times for the entire training dataset. The dataset is processed in batches in order to efficiently compute gradients and update parameters. After each step, old gradients are cleared before computing the new ones. For each image in the batch, a target dictionary is created that includes its bounding boxes and labels. This target information is used to calculate the loss.

5 Evaluation of R-CNN Model Results

At the conclusion of each training period, the model is first assessed on the validation set in order to track performance and identify overfitting. Then, we proceed with the evaluation of the research model.

During this phase, the method is set to evaluation mode to ensure consistent behavior, disabling layers like dropout that are only used during training. For each batch of images from the test set, the model generates predictions, including bounding boxes and labels for detected objects. The process filters out bounding boxes for humans and chairs based on their labels and then checks for

significant overlaps between human and chair bounding boxes.

This overlap detection uses the junction across the union (JoU) metric, and if the IoU exceeds a threshold (0.5 in this case), it indicates that the seat is occupied by a human. The model was shown to have an accuracy of 0.864 and a loss of 0.387 on the set of training. On the set of validation, the accuracy was shown to be 0.853, and the loss was 0.512. The graphs below show the accuracy and loss for each set during the epochs (Figure 7 and Figure 8).

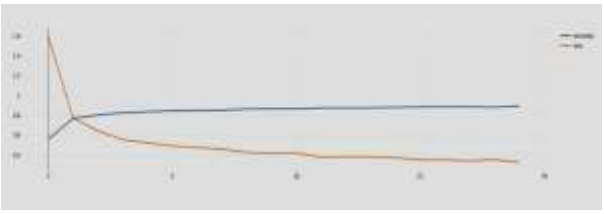

Fig. 7: Evaluation of training and graph of losses

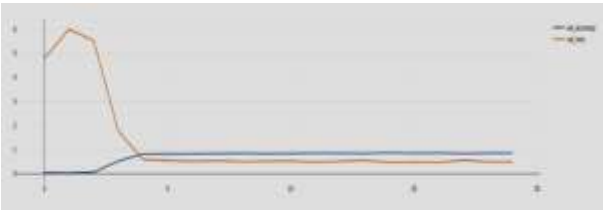

Fig. 8: Accuracy of validation and graph of losses

We demonstrate how Faster R-CNN can be used to show the occupancy of seats in the library. The model's performance, evaluated through both training and testing phases, showed promising results with high accuracy and low loss. Our research performance follows with results of research models, which are summarized in the Table 1.

Table 1. Performance of research models

No. epochs	20
Train accuracy	0.864
Train loss	0.387
Validation accuracy	0.853
Validation loss	0.512

As for the performance evaluation, we refer to Table 2 we use average precision (mAP) as a parameter that can be used to assess the performance of computer vision models. In order to compare many iterations of the same model, it gives a baseline measure that accounts for both precision and variance. We say that is equivalent to the mean of accuracy measurements in our model compressing all the classes.

During the experiments, we go to an evaluation of the mean squared error (MSE) and R^2score, They are calculated for both validation and test sets. The predicted vs actual values are plotted to visualize the results.

Table 2. The performance metrics. The mean square error and R^2 score

	Precision	MSE	R^2Score	AP
predetect	0.68	0.62	0.55	0.16
detection	0.74	0.74	0.58	0.09

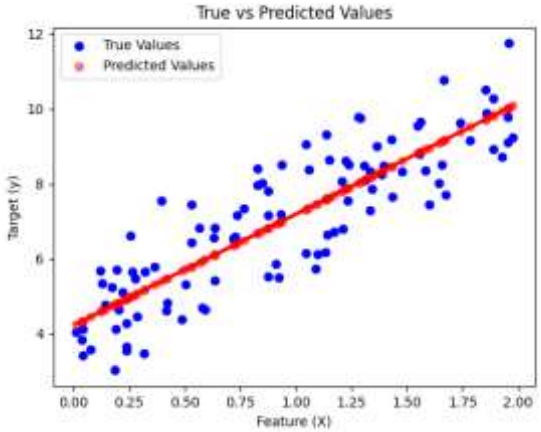

Fig. 9: Linear regression using synthetic data for true and predicted values

What we used from the dataset of all images for each of the algorithms tested follows in the results: test 9.78%, train 78.70%, and validation 19.50%.

6 Conclusion

Our research proposed a polynomial regression model system that leverages the capabilities of Faster R-CNN to develop an automated seat occupancy monitoring system.

The scientific problem performedis fitting a polynomial regression to the data we've generated and visualizing the results. Although the research model was shown to have an overall good performance, there is still a place for improvement. Increasing the size and variety of the training dataset would be a major improvement.

We generate synthetic data using some functions of the different degrees. To perform better, we will use some matrixes of different sizes. Our research gives the result of applying the convolution operation. Its size depends on the input matrix size, and the kernel size.

Our research model would be able to create more reliable features and enhance its capacity to generalize to new, untested data if it were given a wider range of instances from a larger and more diverse dataset. This would most likely increase the

model's dependability and accuracy, particularly in real-world scenarios where circumstances can alter significantly. Graphically, the research reveals the accuracy of validation and graph of losses.

On the other hand, by reducing the dimensionality of the output matrix, the resulting output matrix retains important information about the input matrix. We see that during the convolution process. Convolution is a powerful tool in the processing of images allowing feature extraction and various transformations.

In this article, we investigate the steps involved in preparing a large dataset specifically for the university library and for the methods we used.

During the examination of this process, we conclude that larger data batches can be processed by GPUs, which further improves model performance and training efficiency by providing more reliable gradient estimates. Using GPUs would make the process of developing models more scalable and efficient, allowing for the exploration of larger datasets and more complex designs.

We used original data as for the input matrix like images we wanted to process. Common sizes input matrixes used in our research are 3x3, 5x5, etc. Then, the kernel as a smaller matrix is used to modify the input matrix. It slides over the input matrix to perform all research convolution processes. The result of applying the convolution operation was the output matrix used.

This study lays a solid basis for future research endeavors aimed at seat occupancy recognition by use of object detection techniques. Evaluation of training and a graph of losses was evaluated. We generate data, fit a polynomial regression to it, and visualize the outcome. This is done for all n-splits on both the training and validation sets of each subset of superset as it grows for a bigger dataset on which the model is trained. MSE and R^2 scores are evaluated in our research and are computed for both test and validation sets. To see the outcomes, a plot of the expected and actual values is created in Figure 9.

The methodical approach offered by the well-defined outline of the data processing from loading and pre-processing to model training and evaluation can be expanded upon and improved upon in further research. This work sets the stage for exploring more sophisticated algorithms, refining model architectures, and enhancing the accuracy and efficiency of seat occupancy detection systems.

Declaration of Generative AI and AI-assisted Technologies in the Writing Process

During the last version of this work the authors used Hemingway Editor Service in order to improving the quality of language of article. This doesn't affect the content of article. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

References:

[1] Sharma, N., Jain, V. and Mishra, A. (2018) An Analysis of Convolutional Neural Networks for Image Classification. *Procedia Computer Science*, 132, 377-384. https://doi.org/10.1016/j.procs.2018.05.198.

[2] C. K. Lu, M. W. Mak, R. Li, Z. Chi and H. Fu, "Action Progression Networks for Temporal Action Detection in Videos," in *IEEE Access,* vol. 12, pp. 126829-126844, 2024. https://doi.org/10.1109/ACCESS.2024.3451503.

[3] Li, Zewen; Liu, Fan; Yang, Wenjie; Peng, Shouheng; Zhou, Jun; A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects, *IEEE Transactions on Neural Networks and Learning Systems,* 2022-12, Vol.33 (12), p.6999-7019. https://doi.org/10.1109/TNNLS.2021.3084827

[4] Kubanek M, Bobulski J, Kulawik J. A Method of Speech Coding for Speech Recognition Using a Convolutional Neural Network. *Symmetry*. 2019; 11(9):1185. https://doi.org/10.3390/sym11091185.

[5] Richard, A., & Gall, J. (2016). Temporal Action Detection Using a Statistical Language Model. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 3131-3140. Vegas, Nevada, USA from June 26 to July 1, 2016. https://doi.org/10.1109/CVPR.2016.341.

[6] Xiao Y, Wang X, Zhang P, Meng F, Shao F. Object Detection Based on Faster R-CNN Algorithm with Skip Pooling and Fusion of Contextual Information. *Sensors*. 2020; 20(19):5490. https://doi.org/10.3390/s20195490.

[7] Lin, T., Maire, M., Belongie, S.J., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C.L. (2014). Microsoft COCO: Common Objects in Context. *European Conference on Computer Vision.* Zurich,

Switzerland from September 6 to 12, 2014. https://doi.org/10.1007/978-3-319-10602-1_48.

[8] Gao, J., Yang, Z., Chen, K., Sun, C., & Nevatia, R. (2017). Turn tap: Temporal unit regression network for temporal action proposals. In *Proceedings of the IEEE international conference on computer vision* (pp. 3628-3636). Venice, Italy, from October 22 to 29, 2017, [Online]. https://openaccess.thecvf.com/content_ICCV_2017/papers/Gao_TURN_TAP_Temporal_ICCV_2017_paper.pdf (Accessed Date: December 1, 2024).

[9] Q. Hou, M. -M. Cheng, X. Hu, A. Borji, Z. Tu and P. H. S. Torr, "Deeply Supervised Salient Object Detection with Short Connections," in IEEE *Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 4, pp. 815-828, 1 April 2019, https://doi.org/10.1109/TPAMI.2018.2815688

[10] Wang, L., Qiao, Y., Tang, X., & Van Gool, L. (2016). Actionness estimation using hybrid fully convolutional networks. In Proceedings of the IEEE *Conference on Computer Vision and Pattern Recognition* (pp. 2708-2717). Vegas, Nevada, USA, from June 26 to July 1, 2016. https://doi.org/10.1109/CVPR.2016.296.

[11] Chao, Y. W., Vijayanarasimhan, S., Seybold, B., Ross, D. A., Deng, J., & Sukthankar, R. (2018). Rethinking the faster r-cnn architecture for temporal action localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1130-1139). Salt Lake City, Utah, USA, from June 18 to 22, 2018, [Online]. https://openaccess.thecvf.com/content_cvpr_2018/papers/Chao_Rethinking_the_Faster_CVPR_2018_paper.pdf (Accessed Date: December 1, 2024).

[12] X. Ke, X. Zhang, T. Zhang, J. Shi and S. Wei, "SAR Ship Detection Based on an Improved Faster R-CNN Using Deformable Convolution," *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, Brussels, Belgium, 2021, pp. 3565-3568, https://doi.org/10.1109/IGARSS47720.2021.9554697.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

- Shkelqim Hajrulla carried out the formulation of the problem, simulations and algorithms, optimization process and run codes. He was responsible for the Statistics.
- Özen Özer has implemented the data results, proof reading as corresponding author and carried out all conclusions as for the exact results.
- Taylan Demir has implemented the data results, simulations and plotting the graphics.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The authors have no conflicts of interest to declare.