Automated Attendance Tracking in Classrooms Using YOLOv3

Nandini Ippili

Abstract

In contemporary classroom environments, tracking attendance and gauging student comprehension present significant challenges. Traditional methods such as roll calls and scans are prone to errors. Advances in computer vision and machine learning have revolutionized object detection, with notable algorithms like Viola-Jones and HOG detectors laying the groundwork. However, deep learning, particularly convolutional neural networks (CNNs), has significantly enhanced object detection. This study explores the use of the YOLOv3 model, leveraging the COCO dataset, to automate attendance tracking by modifying it to only detect human figures. Results indicate high confidence levels in most detections, suggesting potential for reliable automated attendance tracking. However, the model's efficacy can be further improved by fine-tuning the dataset and addressing image quality issues. This technology can streamline administrative tasks for teachers, allowing more classroom time to be dedicated to lessons. Today, in the current classroom environment, teachers track attendance through roll calls or scanning the classroom to see which students are absent, which both leave large room for error.

Background

Advancements in computer vision and machine learning have called for massive developments and innovation in the object detection field. Previously, traditional object detection did not use algorithms but rather handcrafted features and shallow neural networks.

One important development during this period was Viola Jones Detectors, which incorporated the concept of “sliding windows” to scale an image in different positions to detect an object (1). This algorithm was meant to be only used for facial detection on black and white photos, although modern object detection was influenced by their techniques (1). The Viola Jones algorithm incorporates “haar wavelets,” which have three types of features that include different sets of black and white rectangles (1). The feature value is the difference between the sum of white pixels and the sum of black pixels (1). For example, in an area where most of the pixels are the same color, the feature value will be little to none, and vice versa (1). A type of data structure called an integral image computes the number of pixels in a rectangular grid. In a grayscale image *I,* the integral image has each of its points (x, y), which contains the sum of all pixels to the left and above (x, y), inclusive (1). In order to compute the integral image, the equation below is used (1). The time complexity for computing the sum of pixels within a rectangular region is o(1). The integral image is computed using Equation 1.

$$ii(x, y) = I(x, y) + ii(x, y - 1) + ii(x - 1, y) - ii(x - 1, y - 1)$$

Fig. 1: Equation 1 (1)

If the base resolution of the detector was 24x24 pixels, there would be around 16000 Haar-like features (1). These Haar-like features are labeled as weak classifiers, as some features are more effective than others (1). Therefore, AdaBoost, a meta-based algorithm, is used to enhance the performance of Viola-Jones algorithm (1).

The Histogram of Oriented Gradient (HOG) detector improves on the Scale-Invariant Feature Transform (SIFT), a technique used to identify features without changing the scaling of an image (5). The HOG detector improves on the Scale-Invariant Feature Transform (SIFT), a technique used to identify features without changing the scaling of an image (4). Similarly to the "sliding windows" concept, the HOG detector utilizes blocks, or pixel grids, that make up the image (5). While the detection window remains the same, the input image is resized to fit this window (5). Then, the gradient of the image is calculated using the magnitude and angle of the image (5). In a block of 3x3 pixels, Gx and Gy are calculated using Equation 2.

$$G_x(r,c) = I(r,c+1) - I(r,c-1) \quad G_y(r,c) = I(r-1,c) - I(r+1,c)$$

Fig. 2: Equation 2 (4)

From the computed Gx and Gy values, the magnitude and angle of the image are obtained from Equation 3.

$$Magnitude(\mu) = \sqrt{G_x^2 + G_y^2} \quad Angle(\theta) = |\tan^{-1}(G_y/G_x)|$$

Fig. 3: Equation 3 (1)

The gradient values from cells, normally 8x8 pixels, which form a block. In each block, a nine-point histogram is computed, depicting the gradient directions in the cell. The histograms combine into one vector, representing the overall appearance of the image.

Techniques such as Viola Jones detectors and HOG detectors involve numerous steps and have many limitations. For example, during gradient backpropagation, a method HOG detectors use, gradients became too small or disappeared completely (6). This era was named pre AlexNet.

A major breakthrough in object detection was the emergence of deep learning and, more importantly, convolutional neural networks (CNN) (6). That marked the era of AlexNet. During this period, the AlexNet model transformed CNN architectures and consisted of five convolutional layers, two normalized layers, two connected layers, three max-pooling layers, and one Soft Max layer (6). Convolutional layers are the building blocks of CNNs and have a set of kernels that extract features (6). Each convolutional layer contained a Rectified Linear Unit (ReLu), an activation function different from other CNNs, which contained a tanh function (6). AlexNet also had Graphics Processing Units (GPU) to enhance training performance, which was not done before (6). Using GPUs, allowed for a shorter training period. The authors utilized data augmentation, mirroring some of the images in the data set, which increased the size of the data (6). AlexNet ended up winning the 2012 ImageNet contest with a top-5 error rate of 15.3%, transforming CNNs into what they are today (6). Since then, numerous advancements have been made in object detection, such as the YOLO model, which defeats human-level accuracy.

Methods

The YOLO (You Only Look Once) model for object detection is developed and evaluated using the methodology illustrated in this section. We will include the dataset and the reasoning behind it, as well as an explanation of the YOLO model architecture and any modifications made to it.

Training

This model is trained on the Common Objects in Context (COCO) dataset, which identifies objects in everyday scenes (7). The model has 2.5 million instances to its 328k images (7). Using a pre-trained model, the main modification made was to only detect humans. Since the project's objective was to track humans, tracking other objects would be precarious for the project. Therefore, I needed to customize the model by adding an additional file of code to only detect humans. I tested the efficacy of the new file by inputting pictures of humans into the model and checking the result.

In order for this model to be implemented in classroom settings, a new dataset of images of the students must be compiled. Each image name must include the corresponding student's first and last name.

Model Architecture

The version of the model used was YOLOv3. This version depends on Darknet-53 as a feature extractor, which uses 53 convolutional layers, a big increase from the previous 19 layers in v2 (2). In this version, it has a higher AP (average precision) or accuracy in detecting small objects, which will prove useful in capturing humans in surveillance cameras (2).

From the feature extractor providing 53 layers, the total convolutional architecture is 106 layers (3). In the 53-layer structure, each layer is followed by a ReLu activation function and a batch normalization layer (8).

The model essentially divides the input image into a grid of S x S cells (3). Each grid cell must predict its own bounding box, confidence score, and class probabilities for the object (8). Since class probabilities are predicted for each cell, there could be multiple bounding boxes for an object (8). However, a post-processing technique, Non-Max Suppression, helps reduce duplicated bounding boxes (8).

Results

Quantitative Results

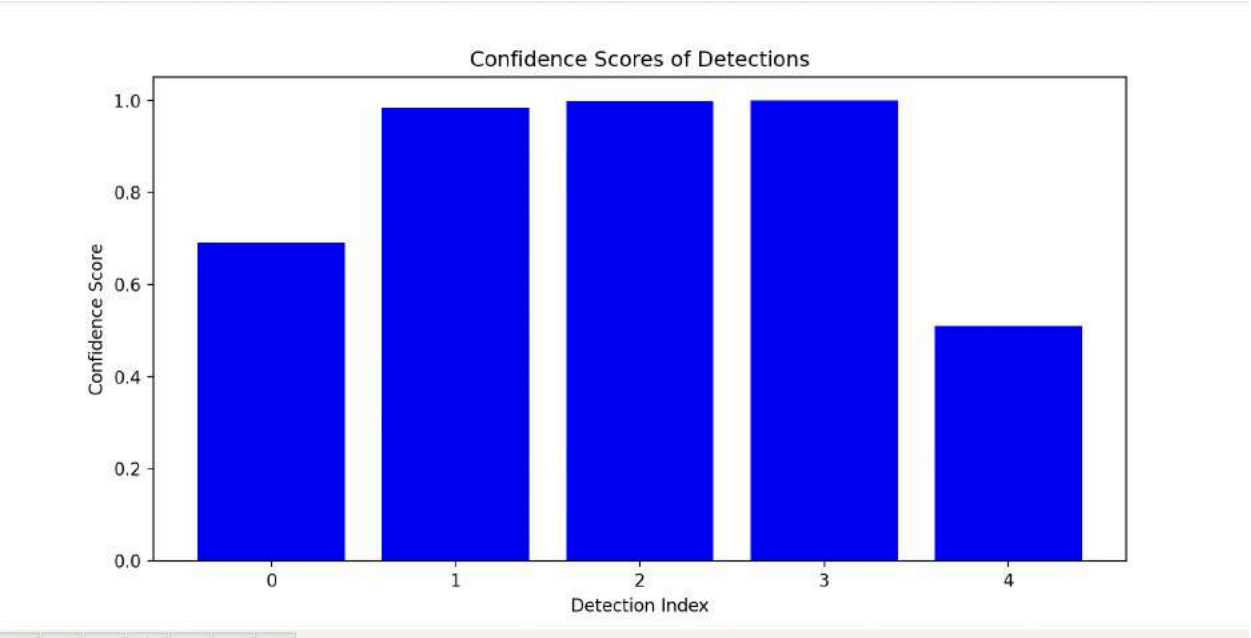

Fig. 4: Confidence Scores of Detections vs. Detection Index

This graph above depicts the confidence scores of the model's object detection plotted against the detection index. Confidence scores show the probability of an image being detected accurately through an algorithm. The detection index is the order of each detected object in the list of all the detections made by the model. The confidence scores range from 0 to 1, where 1 indicates the highest level of confidence. Scores from 0.00-0.50 show a low level of confidence, 0.51-0.70 scores correspond to a moderate level, and finally, 0.71-1.00 recognizes a high level of confidence. At Detection Index 0, the model achieved a score of approximately 0.67, demonstrating a moderate level of confidence. Detection Index 1's confidence score recorded a 0.90, indicating a high level of confidence. Detection Index 2 and 3 achieved the highest

possible score, 1.0, showcasing the model's strong confidence. Detection Index 4 received the lowest score with 0.50, indicating the model's uncertainty with this detection.

Qualitative Results

After inputting this image, the model detects the man as a 'person' and draws a bounding box over the person.

Fig. 5: Man

The model draws bounding boxes over each detected person and no other object, since the model has been altered to only detect humans. Since the people in the image are not in the dataset, the bounding boxes do not have a label for their names.

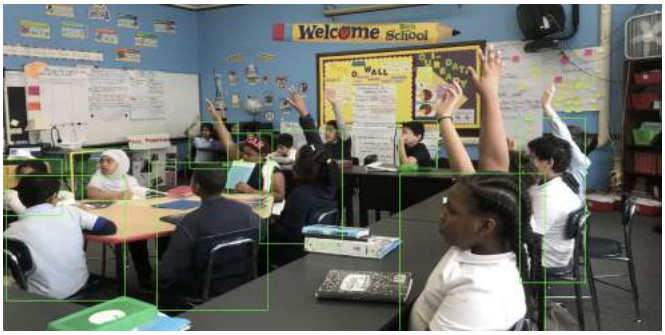

Fig. 6: Students in classroom

Discussion

The results reported from the YOLOv3 model indicate a promising future for automated attendance tracking. By implementing this model, teachers can potentially eliminate traditional attendance tracking methods such as roll calling and manual scanning. With this tool, more classroom time is maximized.

The modified YOLOv3 reported a range of confidence scores across different detections. Detection Index 0 recorded a score of 0.67, which indicated a moderate level of confidence. Future work can improve this detection to increase the model's accuracy and variation. Detection Index 1, 2, and 3, achieved scores ranging from 0.90 to 1.0, ensuring a very high level of confidence in the detections. The scores suggest that the models are highly effective in recognizing human figures entering a classroom. Detection Index 4 reports a 0.50 confidence score, indicating the model's uncertainty in detecting specific objects. Adjusting the quality of the pictures and adding more to the dataset can help improve this detection.

Successful implementation of this model can relieve the unnecessary, administrative burdens teachers have. Increased teaching time can potentially improve students' success in the classroom.

Despite the promising results, certain limitations must be addressed. The YOLOv3 model is trained on the COCO dataset, which may not include that many pictures depicting classroom scenarios. That dataset must be fine-tuned so it can include different images of classroom settings. Poor picture quality could also compromise the results of the model, as it would not be able to clearly draw out features of the image.

However, this modified model does show a promising future for automated attendance tracking. With this tool, classroom time would be extended as teachers would not have to deal with the meticulous task of roll call or screening.

Conclusion

Using the YOLOv3 model for human detection shows great potential for automated attendance tracking. Three detections out of five demonstrate a high level of confidence, indicating the success of the model. The other two detections are opportunities for improvement. With more data adjustment and testing, this technology has the potential to be a crucial component of contemporary learning settings, enabling more precise and effective tracking of student attention and attendance.

Bibliography

1. Aibin, Michael. “The Viola-Jones Algorithm.” *Baeldung on Computer Science*, 18 Mar. 2024, www.baeldung.com/cs/viola-jones-algorithm.
2. Almog, Uri. “Yolo V3 Explained.” *Medium*, Towards Data Science, 13 Oct. 2020, towardsdatascience.com/yolo-v3-explained-ff5b850390f.
3. Chakure, Afroz. “All You Need to Know about Yolo V3 (You Only Look Once).” *DEV Community*, DEV Community, 22 Jan. 2024, dev.to/afrozchakure/all-you-need-to-know-about-yolo-v3-you-only-look-once-e4m.
4. “The Evolution of Object Detection Technologies in AI.” *eNest Technologies*, 6 Mar. 2024, enestit.com/the-evolution-of-object-detection-technologies-in-ai/.
5. “Histogram of Oriented Gradients: An Overview.” *Built In*, builtin.com/articles/histogram-of-oriented-gradients. Accessed 5 Aug. 2024.
6. Klingler, Nico. “Alexnet: A Revolutionary Deep Learning Architecture.” *Viso.Ai*, 19 June 2024, viso.ai/deep-learning/alexnet/#:~:text=Image%20Classification%20%E2%80%93source-,Before%20AlexNet,before%20the%20development%20of%20AlexNet.
7. Meel, Vidushi. “What Is the Coco Dataset? What You Need to Know in 2024.” *Viso.Ai*, 19 June 2024, viso.ai/computer-vision/coco-dataset/#:~:text=The%20COCO%20dataset%20contains%20challenging,of%20real%2Dtime%20object%20detection.
8. Nagpal, Manika. “The Ultimate Guide to Yolo3 Architecture.” *ProjectPro*, 21 Mar. 2024, www.projectpro.io/article/yolov3-architecture/836.