AI-Powered Interactive Educational System for Children Using Computer Vision and Real-Time Recognition Techniques

Zaid Kraitem[1] (Corresponding Author), Dania Al-Tabbaa[2], Tasneem Al-Tayar[3]
Department of Information Engineering, Al-Wataniya Private University (WPU), Hama, Syria[1,2,3]
Email: zaid.kraitem@wpu.edu.sy

***Abstract*—This research presents the design and development of an interactive educational system tailored for children, utilizing artificial intelligence (AI) and computer vision techniques. The system incorporates three core functions: color recognition through the HSV (Hue, Saturation, Value) model, animal detection using the YOLO (You Only Look Once) object detection algorithm, and number recognition via Optical Character Recognition (OCR). These components enable an engaging and responsive learning environment where children interact visually with their surroundings. The system integrates neural networks and real-time image processing to facilitate immediate feedback and personalized learning experiences. Experimental evaluation demonstrated an accuracy of over 93% in color recognition, a mean Average Precision (mAP) of 0.88 in animal detection, and a 91% OCR numbers recognition rate. These promising results indicate the system's potential to significantly enhance early childhood education. This study contributes a practical model combining AI technologies with educational interaction, paving the way for future advancements in AI-assisted learning environments.**

Keywords—Artificial Intelligence, Computer Vision, Interactive Learning, YOLO, HSV, OCR, Neural Networks

1.Introduction

In the digital age, artificial intelligence (AI) and computer vision have become fundamental components of modern technological ecosystems. Their integration into educational settings promises to revolutionize traditional learning methodologies, especially for young children who require dynamic, stimulating environments to foster cognitive and visual development. This paper introduces a novel interactive learning system designed for early learners, leveraging AI to transform passive content consumption into an active, personalized educational experience [1][2].

Artificial Intelligence (AI) and computer vision are transforming the landscape of educational technology. The integration of AI into early childhood learning environments offers new pathways to engage children through visual, tactile, and auditory modes [3][4]. Traditional education methods often lack adaptivity and interactivity, which are critical in capturing children's attention and supporting individualized learning.

This paper introduces an interactive AI system designed for children aged 4–10, enabling real-time recognition of colors, animals, and numbers. By fusing HSV color segmentation, YOLO object detection, and OCR digit interpretation, the system creates a multi-modal learning experience. The research aims not only to design a technically robust framework but also to empirically assess its efficacy in practical learning contexts

The proposed system focuses on three primary capabilities: (1) identifying and categorizing colors in real-world scenes using HSV color space analysis, (2) detecting and naming animals in images and video through YOLO object detection, and (3) recognizing handwritten or printed numbers using OCR technology. These functionalities aim to bridge the gap between tangible objects and digital learning, fostering a multisensory educational environment. As Fig.1 shows.

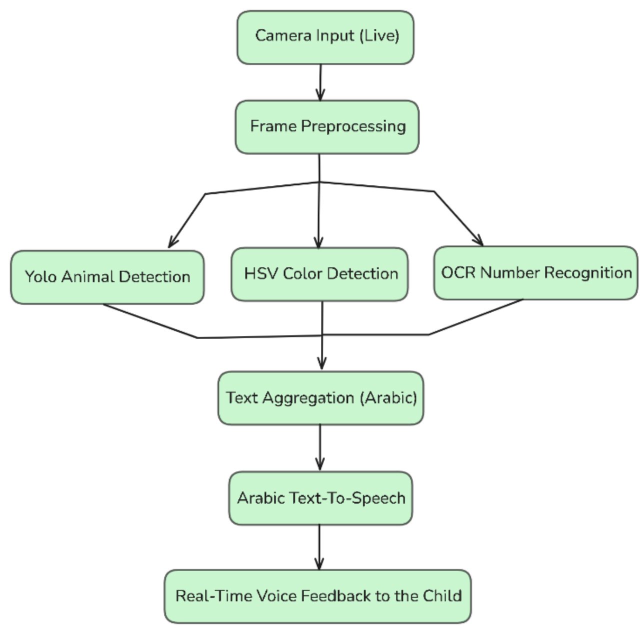

Fig. 1. Flow-Chart of suggested system.

2.Literature Review

2.1 AI and Computer Vision in Education

Previous studies have demonstrated the effectiveness of AI in personalizing learning content and monitoring student behavior [5]. In early learning, where attention span and interactivity are paramount, real-time vision-based systems offer a compelling alternative to static e-learning platforms [6].

Systems such as FINNger [7] have used computer vision to teach basic math using hand gestures, while others have implemented AR/VR modules for immersive learning [8].

We collect our data for animal detection from COCO dataset (100K samples). We add also about 10k images from different resources (Google, Camera). We make augmentation includes Rotation and Flipping to increase the dataset to 300K, and training the model with 80% of these data, and testing it with the remain size.

2.2 YOLO for Real-Time Object Detection

The YOLO (You Only Look Once) family of models represents a significant advancement in object detection. Unlike region proposal methods, YOLO predicts bounding boxes and class labels in a single forward pass [9]. YOLOv5, in particular, balances speed and accuracy, making it suitable for educational environments that require real-time response. Fig2 illustrates the detection steps and the layers of this model.

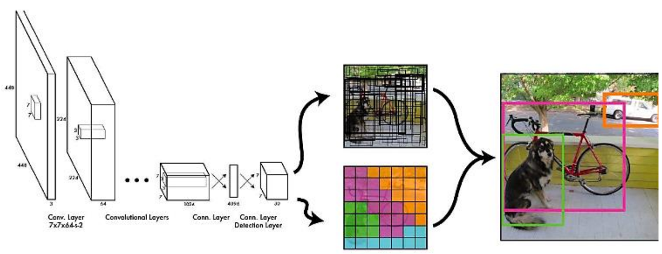

Fig. 2. YOLO Architecture..

2.3 HSV Color Model

While RGB is commonly used in digital imaging, HSV (Hue, Saturation, Value) offers a more intuitive representation of color perception, especially under varying lighting conditions [10]. HSV thresholding has been successfully used in object tracking and segmentation tasks in dynamic settings. Especially since the component expressing lighting is separate in this model (Value component), as the Fig.3 shows.

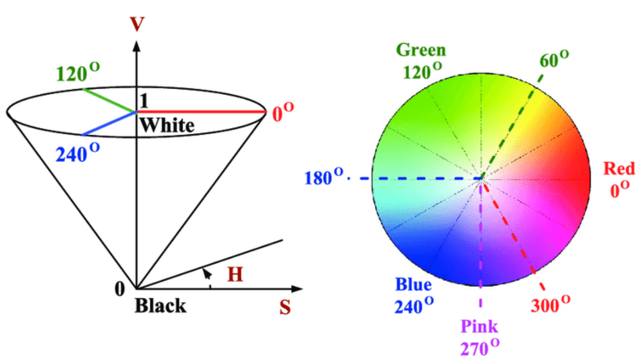

Fig. 3. HSV color model..

2.4 Optical Character Recognition (OCR)

OCR has found applications in document digitization, number plate recognition, and education. Tools like EasyOCR have improved the performance of digit recognition across multiple languages and styles [11]. When integrated into educational tools, OCR allows children to interact with numbers from their physical environment.

First, the input image is preprocessed, cleaned and thresholded to convert it to binary format. Then, a localization stage is performed to isolate the character from the background and accurately determine its boundaries. Then, a segmentation process separates each character separately for final recognition and conversion to the correct corresponding character, as shown in Fig. 4.

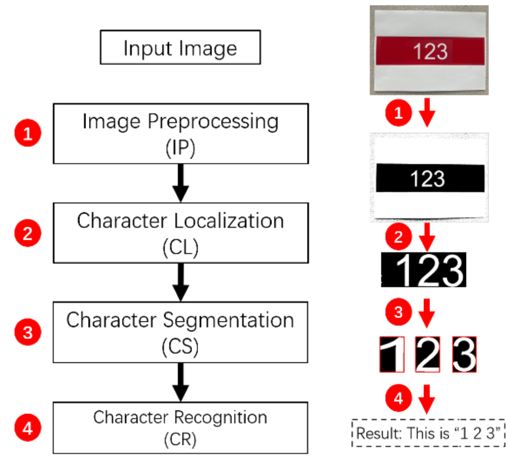

Fig. 4. OCR Stages..

3. System Design and Methodology

3.1 Architecture Overview

The system comprises three visual processing modules implemented on a Python-based platform using OpenCV, PyTorch, and EasyOCR. Input is obtained via a USB or built-in camera, and processed data is relayed back through a graphical interface and audio feedback.

Hardware: Laptop with i5 CPU, integrated webcam, and 8GB RAM Software: YOLOv5 model (Google Collab), EasyOCR, HSV-based OpenCV scripts.

3.2 Module 1: Color Recognition

Colors are detected using HSV thresholding. For instance, red hues are recognized in two ranges due to circular wrapping in HSV space, for example the red color specified as:

Lower red: H=0–10°, S=100–255, V=100–255

Upper red: H=160–180°, S=100–255, V=100–255

In the same way, five colors (red, green, blue, yellow, orange) were evaluated. As Fig. 5 illustrates, the effective of saturation and lightness (value) on the color specification.

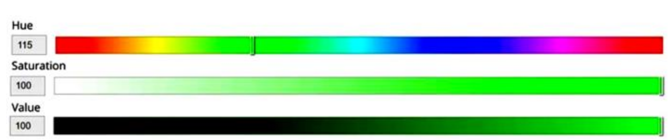

Fig. 5. Determining the range of each color with selecting S and V.

3.3 Module 2: Animal Detection

YOLOv5s was selected for its balance of speed and precision. The training dataset included 10 animal classes (cat, dog, bird, horse, eagle, fish, cow, snake, frog, pig) using the COCO dataset. The model outputs class labels and bounding boxes. Feedback is provided through speech synthesis using pyttsx3. We can calculate (Intersection Over Union IOU) to evaluate the precise of model in object detection through the following equation:

$$\mathrm{IOU} = \frac{A \cap B}{A \cup B} \tag{1}$$

Where A means the anchor box which predicted via the model, while B refer to ideal box which determined before the detection process. If the result was more than 0.7 then the object is detected correctly.

3.4 Module 3: Number Recognition

Digits are recognized from handwritten and printed formats using EasyOCR. The module processes grayscale frames, detects contours, and feeds cropped regions to the OCR engine. We want to recognize the numbers only from 0 to 10. As Fig.6 shows. Note that our model detect number 3 with sound.

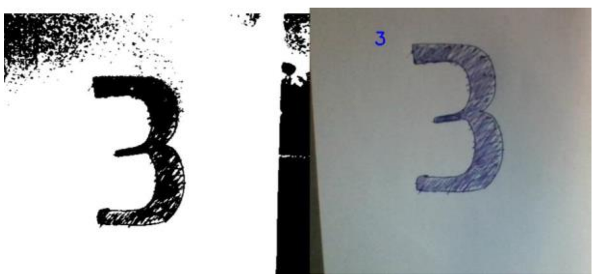

Fig. 6. Left Image: Convert to binary Image, Right Image: OCR recognition result (from our suggested model).

3.5 User Interface

The UI includes large clickable icons, child-friendly themes, and animated transitions. Each function (color, animal, number) is activated via a main menu, with real-time auditory feedback following every successful recognition.

Fig. 7. User Interfaces.

4.1 Experimental Setup and Results

Testing was conducted indoors with varying lighting intensities (300–800 lux). Scenarios simulated home and school settings with real-world objects and cards. For example, in the Fig.8 we test the model using real-time images (Camera) to ensure correct detection using YOLO model.

Fig. 8. Subset of samples for cat detection.

4.2 Metrics

We use several metrics to measure the performance of our three main models and evaluate the system quality in general. Mean Precision (mAP): A metric used to evaluate models in multi-class classification or object detection tasks. It is calculated by:

$$\mathrm{mAP} = \frac{1}{n}\sum_{k=1}^{k=n} AP_k \tag{2}$$

which the AP symbol denotes to average of accuracy for class k, and n represents the number oof classes.

Precision: it measures the percentage of True Positive TP predictions out of the total positive predictions made.

$$\text{Precision} = \frac{TP}{TP+FP} \tag{3}$$

While Recall reflects the model's ability to detect true positive cases. It is calculated as follows:

$$\text{Recall} = \frac{TP}{TP+FN} \tag{4}$$

Response time is measured by the time it takes for a model to respond after receiving input data.

$$Latency = \frac{Response\ Time}{Number\ of\ Input\ Data} \tag{5}$$

If we need to calculate the accuracy for OCR and color detection models, we should test it on a set of numbers. As the following equation:

$$\text{Accuracy} = \frac{C}{T} \times 100 \qquad (6)$$

While C: is Number of correct detections, and T: represents the total number of attempts.

A confusion matrix is used to display classification results in a table format showing correct and incorrect predictions across different categories.in the Fig.9 we note that the confusion matrix in Fig.9 shows that the model correctly classified most images, with 64 images accurately classified as "cat," with very few errors in classification compared to the background.

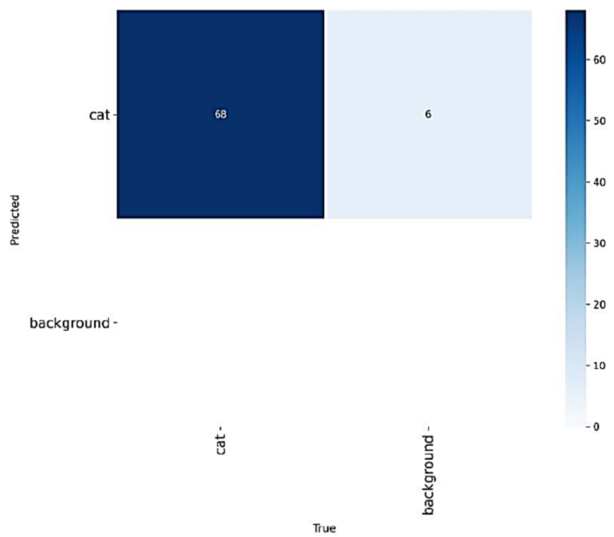

Fig. 9. Confusion Matrix for cat detection.

The F1-confidence curve (Fig.10) illustrates the relationship between the F1 score and the confidence level. The curve shows that the optimal F1 value is at a confidence level of 0.4, where the best balance between precision and recall is achieved.

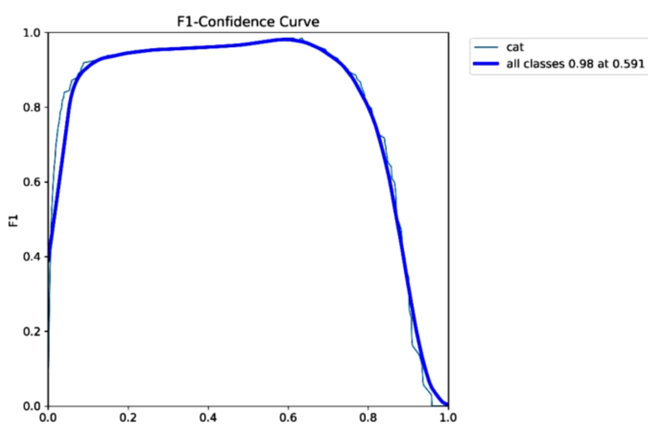

Fig. 10. F1 Score-Confidence curve for cat detection.

We repeat the previous steps and calculate the important parameters to evaluate the performance of the final model for all nine remaining animal species in the same way, as shown in Table I, which includes the most important measurements for cats' detection.

TABLE I. METRICS FOR CAT DETECTION USING YOLO

Metric	Value
F1-Score	0.92
Confidence	0.409
mAP50	0.992
mAP50-95	0.784
Pre-Pocessing Period	16.7 ms
Detection Period	28.4 ms
Post-Processing Period	12.4 ms

We use the same evalution method for OCR. And testing the numbers between 1 and 10, to calculate the accuracy of this model. As Fig.11 and TABLE II show.

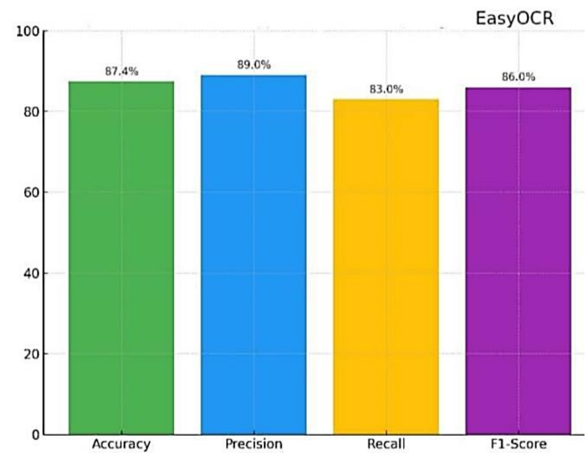

Fig. 11. EasyOCR Performance.

TABLE II. OCR METRICS FOR NUMBERS DETECTION MODEL

Metric	Value
Accuracy	87.4%
Precision	0.89
Recall	0.83
F1-Score	0.86
False Positive	45
False Negative	81

In the color model, A test was conducted to calculate the accuracy of color recognition using a code based on HSV values. The code reads the color of a dot in the center of the screen and then identifies the color name based on these values. However, the code does not automatically calculate accuracy, so a manual test was conducted to calculate the model's accuracy in color recognition. The following steps were followed: The camera displayed known colors, such as red, green, blue, and others. The predicted name printed by the code was then compared with the actual color displayed in front of the camera. If the name displayed was correct, it was recorded as a correct prediction, and if the name was incorrect, it was recorded as an incorrect prediction. These steps were repeated with a set of colors: red, green, blue, yellow, orange, brown, white, black, purple, and pink.

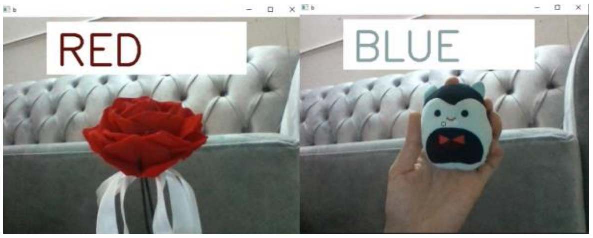

Fig. 12. Color Detection with sound feedback.

The accuracy of color detection was calculated after 40 attempts and reached 93.4%. The reason for the failure of prediction in some cases was due to several factors, including: lighting - similarity of the color range (such as brown and orange, for example) - shadows and reflections - proximity and distance from the camera.

4.3 Error Analysis

HSV color misclassifications were caused by backlight reflections and shadows. YOLO misidentified overlapping animals (e.g., dogs and wolves). OCR struggled with child-written digits "3" and "5" due to loop similarity

5. Discussion

The results confirm the system's robustness in recognizing visual stimuli under real-world conditions. Its modularity and immediate feedback make it well-suited for inclusion in early educational curriculums. Compared to related works [5][6], our system uniquely integrates three recognition tasks in real time, offering a more holistic learning experience.

TABLE III. TABLE TYPE STYLES

Module	Metric	Result
Color Recognition	Accuracy	93.4%
Animal Detection	mAP@0.5 (YOLOv5)	0.88
OCR Digits	Recognition Rate	91.2%
Animal Detection	Precision / Recall (avg)	0.85 / 0.86

However, limitations persist. The HSV model is sensitive to ambient lighting; dynamic HSV adaptation may improve performance. YOLOv5 requires fine-tuning for child-drawn animals or fantasy illustrations. OCR modules could be enhanced using CNN-based digit classifiers instead of generic OCR engines.

6. Conclusion

This research demonstrates a successful implementation of an AI-based learning system for children, integrating HSV color detection, YOLO object identification, and OCR-based numeric interpretation into a unified platform. The system achieves high recognition rates and engages learners through visual and auditory interaction. Future work includes deploying the system in real classrooms, incorporating adaptive learning paths, and expanding to language recognition modules.

7. References

[1] Luckin, R., et al. (2016). Intelligence Unleashed: An Argument for AI in Education. Pearson Education.

[2] Woolf, B. (2010). Building Intelligent Interactive Tutors. Morgan Kaufmann.

[3] D'Mello, S., & Graesser, A. (2012). AutoTutor and Affective Learning. IEEE Transactions on Learning Technologies, 5(1), 74–87.

[4] Blackwell, A. F., & Oliver, M. (2019). Children's Interactions with AI Systems. Learning Media and Technology, 44(2), 123–135.

[5] Oduibert, V., & Mascio, R. (2021). FINNger: AI Application for Learning Math via Finger Counting. Journal of AI and Education, 3(2), 88–97.

[6] Zhang, X., et al. (2018). Interactive AR Systems in Primary Education. ACM SIGCSE.

[7] Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. arXiv:1804.02767

[8] Patel, A. (2020). Understanding HSV Color Space in OpenCV Python. Towards Data Science.

[9] Smith, R. (2007). An Overview of the Tesseract OCR Engine. ICDAR.

[10] Lin, T.Y., et al. (2014). Microsoft COCO: Common Objects in Context. ECCV.