

ZYNGA SUBMISSION

Team Name: VisionQuest

PROBLEM STATEMENT 1 :

For Sets 1 to 7, Develop an AI / ML model to identify the main image from two test images within a set. The varying resolutions of the test images add complexity to this task. The algorithm should return the similarity percentage of the main image relative to its matching counterpart in the test images. Return the similarity percentages in the following format: [percentage_for_test1, percentage_for_test2].

SOLUTION:

- We developed a Machine Learning model to compare a “main image” against two test images in each set (provided dataset). We aimed to determine similarity percentages between the main image and each test image.
- We designed an algorithm to handle varying resolutions of test images, ensuring robust performance across different image qualities.
- We have built a **Convolutional Neural Network (CNN)** model to extract features from images. The CNN consists of **two convolutional layers** followed by **max pooling** to reduce spatial dimensions and enhance feature extraction.
- We have calculated **cosine similarity scores** between the feature vectors of the main image and two test image and employed cosine similarity metric to compare feature vectors extracted from the main image and each test image since **cosine similarity provides a measure of similarity ranging from 0 (completely dissimilar) to 1 (identical)**.
- For the execution of our model, the user needs to enter the paths of the files (Main, Test1, and Test2). The system loads these images, computes their respective feature vectors using the CNN model, and calculates similarity percentages and the output will be displayed on the screen.

PROBLEM STATEMENT 2 :

For Set 8, Extend upon the same AI / ML model solution to identify the total win amount. In the image, it will be found in the TOTAL WIN box. Return the total win amount in the following format: [total_win_amount_test1, total_win_amount_test2].

SOLUTION:

- We built a Convolutional Neural Network (CNN) to extract features from images. The CNN model includes layers of convolution, max pooling, and fully connected layers to capture and flatten image features. This model is used for image similarity tasks.\
 - For Set 8, we specifically focused on extracting the total win amount using Optical Character Recognition.
 - We initialized the reader with English language settings to detect text regions within the images.
 - A function **extract_total_win_amount** is implemented to identify and extract the total win amount specifically from the "TOTAL WIN" box in the images. This function scans the detected text regions and extracts numerical values associated with "TOTAL WIN".
-

PROBLEM STATEMENT 3:

For Set 9, Return the bet amount present in the BET section in the image. Return the bet amount in the following format: [bet_amount_test1, bet_amount_test2]

SOLUTION:

- Since in this problem , we need to find the “BET” amount , therefore we labelled the Images by finding the desired **Region of Interest (ROI)**. We Used green rectangles(box) for Numbers and Red rectangles(box) for Text.
- For effective classification we created a dataset by playing game and then annotated them by defining bbox_distance (we created an algorithm to find the nearest number situated near to “BET” text).
- A function named **bbox_distance** is defined to calculate the **Euclidean distance** between the centers of two bounding boxes. As the code iterates through the detected text, it checks if the detected text matches "BET" and stores its bounding box if found. A green rectangle is drawn around text containing digits, and its distance to the "BET" bounding box is calculated. The code tracks the closest numerical value to "BET" by comparing these distances. Text without digits is enclosed in a red rectangle.
- We created a CNN model and trained it on our Own created dataset. The image is read using OpenCV, and text detection is performed using OCR. Detected text along with their bounding boxes and confidence scores are obtained. A threshold is set to filter out low-confidence detections.

- Finally, the image with the highlighted bounding boxes is displayed using Matplotlib, and the closest numerical value to "BET" is printed. This method effectively combines OCR for text detection and spatial analysis to find and highlight relevant numerical information in the image



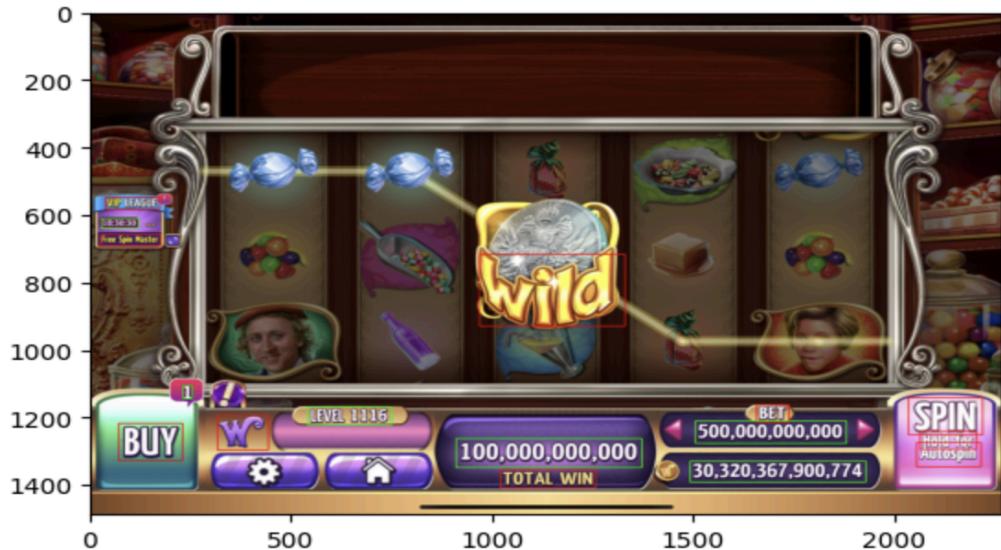
Before: Test.png



After: Test1.png

Some of our outputs are shown below :

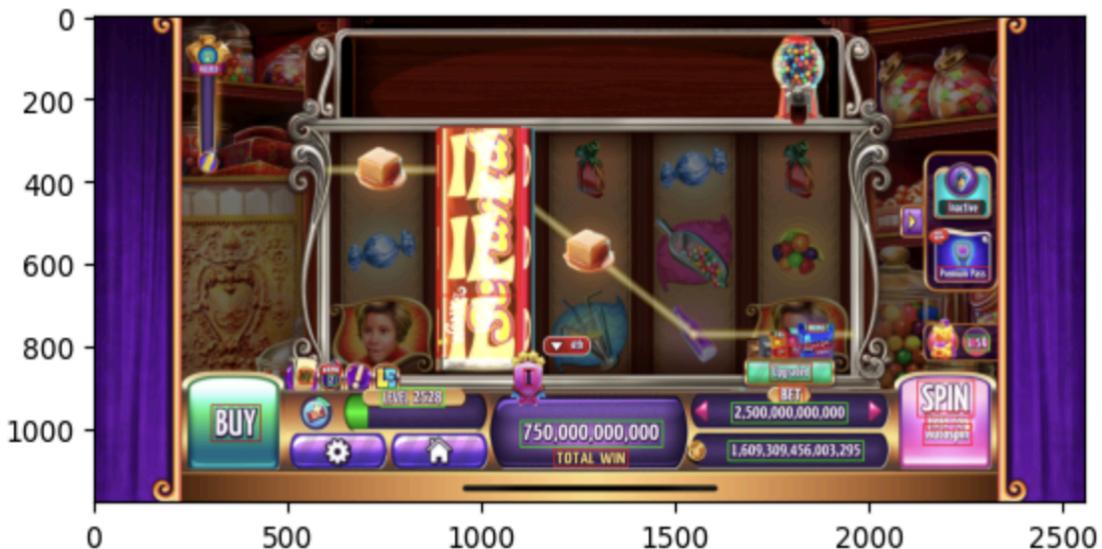
- WARNING:easyocr.easyocr:Using CPU. Note: This module is much faster with a GPU.



The closest numerical value to 'BET' in Test5.png is: 500,000,000,000



The closest numerical value to 'BET' in Test25.jpg is: 15,000,000



The closest numerical value to 'BET' in Test6.png is: 2,500,000,000,000



The closest numerical value to 'BET' in Test19.jpg is: 7,500,000