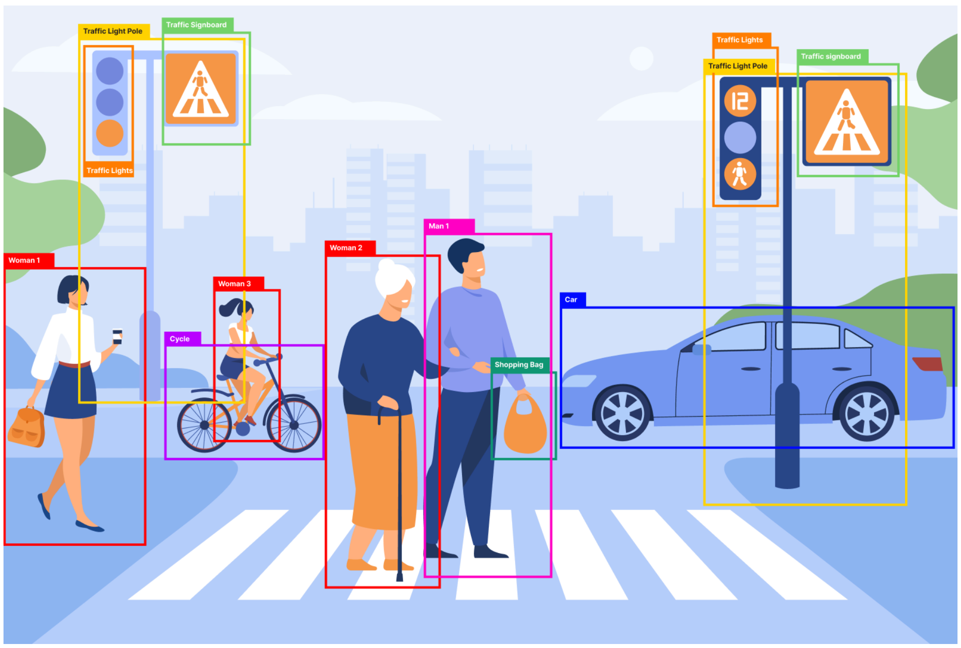
AI Dataset Vs Object Detection Accuracy & Speed



June 6, 2025

Abstract

This project examines the impact of dataset size on the accuracy and speed of AI object detection models. AI object detection is used in robotics and self-driving cars to identify objects in images or videos. The question of this experiment is does increasing the dataset size improves the model's performance in detecting objects accurately and efficiently. To explore this, three AI models were trained using three different dataset sizes: small(10 images), medium (100 images), and large(1000 images). Each model was evaluated based on two key variables: detection accuracy and processing speed. The training and testing were done using TensorFlow in Google Colab without manual data labeling. The results showed that the model trained on the medium dataset achieved 7% higher accuracy than the one trained on the highest dataset, and a 55 % higher accuracy than the small datasets. And, the medium dataset also had a faster processing speed compared to the small and large datasets by 0.01 seconds. These findings highlight the importance of balancing dataset size with strict data management and training time when developing AI systems. Larger datasets generally enhance accuracy but require more time and power to process. Understanding this balance can help developers optimize AI systems for real-world applications like security systems or robots, where both accuracy and efficiency are important. Overall, the project successfully demonstrated how dataset size impacts AI object detection performance and provided useful information for improving future AI development.

Background Research

Artificial Intelligence (AI) is a new technology that allows machines to think, learn, and act like humans by analyzing data and patterns (Stryker, 2024). AI has recently become very popular, having been used in many applications, such as text or image generation. Platforms like ChatGPT have reported over **400.61 million** monthly active users in February 2025 (Alves 2024). AI is also very popular with businesses; Forbes says **72% of businesses** have adopted AI for at least one business function (Haan 2024). AI is also used for **object detection**, which is the focus of this experiment. AI object detection is a technology that combines **machine learning** and **computer vision** (images/videos) to identify and classify objects within images or videos (Murrel, 2024). Object detection's first component is **Object Localization**, which finds the object's position by creating imaginary boxes around the object. The second component of object detection is **Object Classification**, which categorizes objects into specific groups. The purpose of this experiment is to find out how the difference in AI training data affects the accuracy and speed of object detection.

The science behind object detection uses Artificial intelligence, machine learning, **deep learning**, and **Convolutional Neural Networks (CNN)**. The training of AI is called machine learning, which is training models or systems to make predictions or decisions based on data. Machine learning allows computers to learn from data and patterns with no human interference (Stryker, 2024). There are many types of machine learning algorithms/systems for training AI, however, the most popular is **neural networks**. Neural networks are designed after human neural networks, there are many layers connected that work to process and analyze data (Yasar, 2024). The structure of the neural network is: the input of information, the hidden layers that process information, and the output, which gives the results. Neural networks are very effective in solving complex problems and patterns in large datasets. The most advanced version of machine learning and neural networks is deep learning. Regular neural networks have one to two hidden layers, but deep learning has hundreds of hidden layers, making it capable of solving complex patterns like computer vision. Deep learning plays a big role in object detection because it can analyze and identify visual images or videos, which are very complex data, and produce an output (LeCun, 2015). In object detection, the output would be finding the position of the object and categorizing it into a group.

Convolutional Neural Networks (CNNs) are a specific deep learning algorithm that is used in object detection (Gillis, 2024). CNNs are specialized for receiving and analyzing data from images/videos, as it is very effective in image recognition & analysis, and object detection. CNNs can automatically process data and produce an output, on the other hand, normal neural networks have to do it step by step, which is very slow. CNNs have a structure called a feature map, where the current layer sets the foundation for the next layer, and there are multiple layers in CNNs. The first layer is the **convolution layer,** in which kernels (filters) detect features like edges and texture. Kernels can detect features because they measure the pixels in images, which are tiny. The kernels are weights that measure the pixel value and produce the output value. The output value is sent to the feature map, where it will be used in the next layer. There are also many convolution layers in CNNs; the earlier layers measure simple lines, and the deeper layers measure complex lines. The second layer is the **pooling layer,** which uses kernels again, but this time, instead of getting pixel values, the kernels remove pixel values that it does not need. Pixel values that are important and of high value remain, while those with low value are removed. This is done to make the image size smaller and processing much easier and efficient. The final and most important layer is the **fully connected layer,** which makes the final classification by combining the data it received from the first two layers and putting them into different classes. There are also additional layers, like the **activation function,** that can analyze non-linearity (complex non-straight lines), allowing the AI to analyze more complex features and shapes.

This experiment utilizes TensorFlow, a popular machine learning software framework. TensorFlow is popular for industrial applications, with over 30,000 public companies using it, including major companies like Google, Amazon, and General Motors. The most user-friendly and effective machine learning model in Tensorflow is the Keras Sequential Class. The Keras Sequential Class is ideal for building neural networks and CNNs because it allows developers to build models layer by layer, like stacking blocks. To build a CNN model using Keras Sequential, **Conv2D()** is used first to add a convolution layer. This is followed by **MaxPooling2D()** to add a pooling layer**.** The **Flatten()** function is then used to convert the outputs from 2D into a 1D list so it can be passed to the fully connected layer. The **Dense()** function is used to add a fully connected layer where each neuron calculates its output by using the mathematical formula (output = activation (dot(input, kernel) + bias)). In this formula, the input is the data received from the previous layer, a **kernel**(weights) is used to find the value of each pixel, the **bias** shifts the result to help the model learn better, the **dot** is used to multiply the input and add it up, and **activation function** decides whether the neuron should activate or be zero, which adds non-linearity (complex shapes) to the model. After building the layers, the model is compiled or configured with the settings for the training. This step configures the training process by setting the **optimizer,** which is the algorithm that updates the model's kernel; the **loss function**, which measures how far off the predictions are from the true labels; and metrics like accuracy, which help evaluate the model's performance. Once compiled, the model is trained using the **fit()** function, which uses input data (train_x_rgb) and correct answers (train_y) for the input data. The model is trained over a number of **epochs**, which means how many times the training data will be reviewed. It also uses a **batch size**, which is the number of data points processed before the model's values change, and a **validation split**, which is the testing images that the model will test itself on. The parameters epochs, batch size, and validation greatly impact the overall accuracy of the model, so throughout the experiment, they will be kept the same for optimal results.

The efficiency and result of the AI itself depend on the quality & quantity of training data, or simply put, the training process itself. For AI to learn, it must be given information, which can be in the form of text, visual data, or sensor data (Jaen, 2024). After information has been given, there are two ways AI can be trained: supervised **learning** and **unsupervised learning**. In supervised learning, a human must manually label data to guide AI to identify patterns. In unsupervised learning, the AI finds patterns without human interference, allowing it to solve unknown problems. This experiment will focus on supervised learning, where the training images are manually labeled by organizing them into folders (e.g., Apple and Banana), allowing the model to learn from known categories. The training process involves multiple steps. The first step in the training process is **data collection,** where a large amount of formatted data (text, image, etc.) is collected and stored. The second step in the training process is **data labeling,** in this case, done by sorting images into category-specific folders. The third step in the training process is **data validation,** where errors and the quality of data are checked. The fourth step is **data pre-processing,** in which the data is organized and cleaned to ensure better efficiency in the training. The final step in the training process is the training itself. Here, the AI uses the labeled data from the human to learn specific patterns, similar to how students learn from teachers. After that, the AI is tested with new images, repeatedly learning more patterns and improving its efficiency. This experiment will focus on the data collection and labeling portion of the AI training process.

AI and object detection provide benefits for businesses and individuals. A study by PwC estimates that AI technology will contribute **$15.7 trillion** to the global economy by 2030, and boost local economies' GDP by **26%**. However, AI and object detection also have countless downsides. Google, Microsoft, Meta, and OpenAI used about **48 billion liters** of water for their data centers in 2023 (Alaves 2024). AI's water usage could hit up to **660 billion liters** by 2027, which is about half as much water as the UK uses in a year (Hughes 2024). Colombia Climate School revealed AI accounts for **2.5 to 3.7 percent** of global greenhouse gas emissions and uses **0.9 to 1.3 percent** of global electricity demand, which is estimated to increase to **1.85 percent** by 2028 (Cho 2023). These statistics show that AI's increasing usage is significantly impacting the environment. Therefore, finding optimal frameworks for training AI is crucial for efficiency and protecting the environment.

Purpose

The purpose of this experiment is to determine how the quantity of AI training data affects the accuracy and speed of object detection. The independent variable is the number of training images (low: 10 images, medium: 100 images, high: 1000 images), and the dependent variable is the AI's accuracy (confidence level and correct predictions) and speed (time taken to detect objects in milliseconds).

Hypothesis

If the amount of AI training data increases, then the accuracy and speed of object detection will improve because a larger dataset allows the AI to learn more patterns, leading to better object recognition and faster processing.

Materials

- Computer
- Image Folder (With 4 Sub-Folders)
  - High Image Folder (1000 images)
  - Medium Image Folder (100 images)
  - Low Image Folder (10 images)
  - Test Image Folder (40 images)
- Softwares Needed
  - Google Chrome
  - Google Sheets
  - Google Colab

Procedure

1. Find and download 575 images of bananas and 575 of apples (total: 1150 images).
2. Create a main folder named "Dataset" with four subfolders:
    a. Low Images (10 total: 5 bananas, 5 apples)
    b. Medium Images (100 total: 50 bananas, 50 apples)
    c. High Images (1000 total: 500 bananas, 500 apples)
    d. Test Images (40 total: 20 bananas, 20 apples)
3. Upload the entire Dataset folder to Google Drive for use in Google Colab.
4. Open Google Colab in a web browser and create a new notebook.
5. Mount Google Drive in Colab using the command:

```
from google.colab import drive
drive.mount('/content/drive')
```

6. Set paths to each dataset within the mounted Google Drive.

```
base_path = "/content/drive/MyDrive/ScienceFair"
low_path = f"{base_path}/lowimages"
mid_path = f"{base_path}/midimages"
high_path = f"{base_path}/highimages"
test_path = f"{base_path}/test"
```

7. Load the datasets using TensorFlow's image_dataset_from_directory() function.
8. Create a Convolution Neural Network model using TensorFlow's Keras API. It must include the following layers:
    a. Rescaling
    b. Conv2D
    c. MaxPooling2D
    d. Flatten
    e. Dense (128 units, ReLU)
    f. Dense (2 units, output layer for apple and banana)

9. Compile the model using an optimizer, list the loss function and accuracy as the metric.
10. Train three models for 15 epochs each using the low, mid, and high datasets, respectively.
11. Load the test dataset using the same steps as the three models for predictions.
12. Evaluate each model on the same test dataset using a loop that:
    a. Records the prediction label and actual label.
    b. Measures the time taken to predict.
    c. Calculates whether the prediction was correct.
    d. Stores prediction results for each image in a list
13. Calculate the accuracy and average prediction time for each model.
14. Write down the prediction in a table with the following format:

Test	Data	Small Dataset	Medium Dataset	Large Dataset
Test 1	Correct Predictions			
	Wrong Predictions			
	Accuracy (%)			
	Speed (s)			

15. Repeat steps (10-14) three times by retraining and evaluating the model. Record findings in the chart.

Results

The results from the 37 trials involving almost 5000 object detection tests show a pattern between the training dataset and the model accuracy. The Medium Dataset (MD), performed the best with an average accuracy of 91.9%, with a processing speed of 0.14 seconds. The Large Dataset (LD) also performed well with an average accuracy of 84.5%, but it had a slower processing speed compared to the MD. The Small Dataset (SD) had the lowest accuracy at 37.2%, even though its processing speed was the same as the LD at 0.15 seconds. The results show that while increasing dataset size improves performance, the MD provided the best balance between accuracy and efficiency. The LD lower accuracy compared to MD is due to overfitting, a common issue in deep learning which is due to unbalanced, large unregulated data. A possible source of error in this experiment was image quality inconsistency such as lightning, resolution, angle variation, could have skewed the learning process. Therefore, the data suggests a well-balanced, decent-sized dataset is optimal for training AI object detection models.

Test	Data	Small Dataset	Medium Dataset	Large Dataset
Total	**Correct Predictions**	550	1360	1251
	Wrong Predictions	930	120	229
Average	**Accuracy (%)**	37.2	91.9	84.5
	Speed (s)	0.15	0.14	0.15

Conclusion

The purpose of this project was to determine how the size of AI training data affects the accuracy and speed of object detection. The hypothesis that if the amount of AI training data increased, then the accuracy and speed will also improve. The results partially supported the hypothesis. The results demonstrated that the Medium Dataset (100 images) delivered the best overall performance, with an average accuracy of 91.9% and a processing speed of 0.14 seconds. In comparison, the Small Dataset (10 images) and the Large Dataset (1000 images) both had slower performance: the Small Dataset achieved only 37.2% accuracy, and although the Large Dataset reached 84.5% accuracy, it did not surpass the Medium Dataset and had a similar processing speed of 0.15 seconds. While increasing the dataset size from small to medium improved the model's performance, the Large Dataset did not result in significant accuracy gains and even showed a slight decrease in accuracy. In this experiment, the Large Dataset was not managed using techniques like early stopping or validation checks, allowing the model to train without guidance, which may have led to overfitting or reduced efficiency. This highlights a relationship between the independent variable (dataset size) and the dependent variables (accuracy and speed), indicating that there is an optimal data range beyond which performance may plateau or decline without proper data management. For future experiments, an improvement would be to get better quality images for consistent results. Future research would be exploring advanced data management strategies and training techniques that can help identify the best conditions for training object detection AI. This experiment successfully identified the most efficient dataset size range, offering insights that could improve the development of optimal AI systems. These systems can help reduce AI's environmental impact while also creating efficient object detection systems for autonomous vehicles and robotics.

References

Alves, B. (2024, November 22). *Environmental impact of AI - statistics & facts*. Statista. Retrieved June 2, 2025, from https://www.statista.com/topics/12959/environmental-impact-of-ai/#topicOverview

Cho, R. (2023, June 9). *AI's Growing Carbon Footprint – State of the Planet*. State of the Planet. Retrieved June 2, 2025, from https://news.climate.columbia.edu/2023/06/09/ais-growing-carbon-footprint/

Gillis, A. S. (2024, November 25). *What is a Convolutional Neural Network (CNN)? | Definition from TechTarget*. TechTarget. Retrieved June 4, 2025, from https://www.techtarget.com/searchenterpriseai/definition/convolutional-neural-network

Haan, K., Editor, S., & Holznienkemper, L. (2024, October 16). *22 Top AI Statistics & Trends – Forbes Advisor*. Forbes. Retrieved June 2, 2025, from https://www.forbes.com/advisor/business/ai-statistics/

Jaen, N. (2024, March 26). *How AI is trained: the critical role of training data – RWS*. RWS. Retrieved June 4, 2025, from https://www.rws.com/artificial-intelligence/train-ai-data-services/blog/how-ai-is-trained-the-critical-role-of-ai-training-data/

Jain, S. (2024, September 17). *Keras Sequential Class*. GeeksforGeeks. Retrieved June 2, 2025, from https://www.geeksforgeeks.org/keras-sequential-class/

Keras documentation: Dense layer. (n.d.). Keras. Retrieved June 2, 2025, from https://keras.io/api/layers/core_layers/dense/

Keras documentation: Flatten layer. (n.d.). Keras. Retrieved June 2, 2025, from https://keras.io/api/layers/reshaping_layers/flatten/

LeCun, Y. (2015, May 27). *Deep learning*. Nature News. https://www.nature.com/articles/nature14539

PwC. (n.d.). *PwC's Global Artificial Intelligence Study: Exploiting the AI Revolution*. Artificial Intelligence. https://www.pwc.com/gx/en/issues/artificial-intelligence/publications/artificial-intelligence-study.html?theme=yellow&__hstc=233546881.d24881d106468e94c86cb34531a3b213.1704466257685.1704466257685.1704466257685.1&__hssc=233546881.103.1704466257686&__hsfp=24

TheirStack. (n.d.). *Companies that use TensorFlow*. Data Science And Machine Learning. https://theirstack.com/en/technology/tensorflow

What Is Artificial Intelligence (AI)? (2024, August 9). IBM. Retrieved June 4, 2025, from https://www.ibm.com/think/topics/artificial-intelligence

What is Object Detection? (n.d.). IBM. Retrieved June 4, 2025, from https://www.ibm.com/think/topics/object-detection

Yasar, K. (2024, September 18). *What is a Neural Network? | Definition from TechTarget*. TechTarget. Retrieved June 4, 2025, from https://www.techtarget.com/searchenterpriseai/definition/neural-network