

# Visual Recognition with Convolutional Neural Networks for Object Detection

Mohit Bhosale<sup>1</sup>, S.K. Shinde<sup>2\*</sup>, Nikita Bhosale<sup>3</sup>, Tejesh Allapure<sup>4</sup>

## Abstract

Various research and development have taken place over the years on computer vision which is a branch of artificial intelligence (AI). AI disciplines like a vision system is applied in various fields like self-driving cars, face detection by social media apps and law enforcement software's Google Lens and so on. The proposed system deals with design and implementation of an efficient way of training a graphics processing unit (GPU) using Python libraries to process and classify an image. The design is profoundly described along with its performance efficiency. The image processing techniques presented in this classifier includes the concept of neural networks and machine learning. The base for both model training and model evaluation is provided by Cifar-10 dataset. The core of the image classification system consists of deep neural networks, specifically convolutional neural networks (CNNs) linear support vector classification (SVC). The results highlight the model's potential uses in a variety of fields, including healthcare, autonomous vehicles, and content recommendation systems, and show how well the model performs when accurately categorizing a range of photos. The results highlight the model's potential applications in various fields, such as healthcare, autonomous vehicles, and content recommendation systems. They demonstrate the model's proficiency in accurately categorizing a wide range of images. This system not only showcases the advancements in computer vision but also underscores the practical implications of these technologies. By leveraging CNNs and other machine learning techniques, the proposed system achieves high accuracy and efficiency, making it a valuable tool for real-world applications. In conclusion, the design and implementation of this image classification system represent a significant step forward in the application of AI in image processing, offering robust solutions for diverse industries.

**Keywords:** Data set, model selection, training, monitoring and maintenance, support vector classification (SVC), convolutional neural network (CNN), Python

### \*Author for Correspondence

S.K. Shinde  
E-mail: santoshk.shinde@gmail.com

<sup>1,3,4</sup>Student, Department of Electronics & Telecommunication, Smt. Kashibai Navale College of Engineering, Vadgaon, Pune, Maharashtra, India

<sup>2</sup>Professor, Department of Electronics & Telecommunication, Smt. Kashibai Navale College of Engineering, Vadgaon, Pune, Maharashtra, India

Received Date: July 16, 2024,  
Accepted Date: July 29, 2024  
Published Date: August 7, 2024

**Citation:** Mohit Bhosale, S.K. Shinde, Nikita Bhosale, Tejesh Allapure. Visual Recognition with Convolutional Neural Networks for Object Detection. Trends in Opto-electro & Optical Communication. 2024; 14(2): 7–13p.

## INTRODUCTION

Artificial intelligence (AI) has become an essential part of human life, transforming various industries with its capabilities. One significant branch of AI is computer vision, which focuses on enabling machines to interpret and process visual information from the world. Vision sensing systems are the primary focus of this project. This system aims to train a deep learning model on a dataset of images and subsequently use the trained model to classify images. Such a classifier system can identify various types of images, such as recognizing a dog breed, identifying a car model from its picture, or determining a plant's name from a photo of its flower. For instance, our application

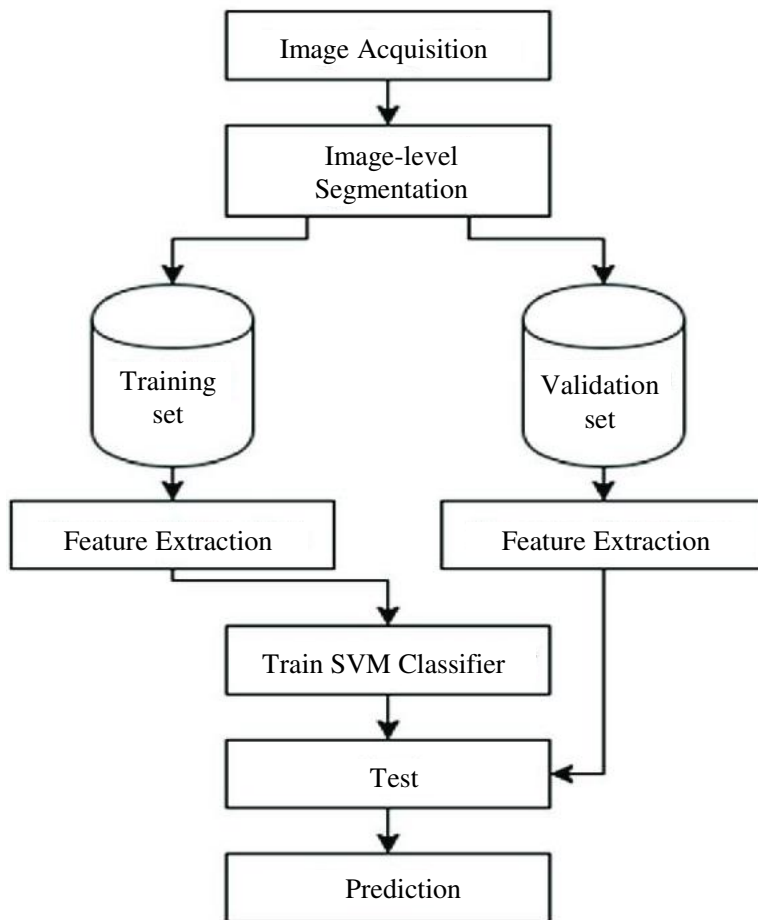
will perform the latter task. Code is developed in PyCharm to ensure the training implementation works, and it is subsequently executed from the command line. In our application, Python library packages such as OpenCV, Tkinter, and Matplotlib are utilized. Image classification is a supervised machine learning task that involves training a model to identify and classify objects in images. These models are used in a wide range of applications, such as medical imaging, product recognition, self-driving cars, and satellite imagery analysis. The advancements in deep learning, particularly in convolutional neural networks (CNNs), have significantly improved the accuracy and efficiency of image classification tasks.

## PROPOSED SYSTEM

In the image acquisition block, the image is sourced either from the internet or from the camera. This system aims to automatically classify images into predefined categories based on a trained machine learning model. The system will have a user-friendly interface for users to upload new images for classification. The image segregation block helps in separating the images into two pools, namely, testing set and the validation set. At the core of the system is a pre-trained machine learning model for image classification. This could be a deep learning model (CNN) or a traditional machine learning model depending on the complexity of the task. The system will automatically preprocess uploaded images to ensure consistency with the training data. This involves feeding the images and their corresponding labels to the algorithm and adjusting the model's internal parameters to minimize classification errors. The testing set is used for training the model while the validation pool is used for testing and predicting. Machine learning models can achieve high accuracy in image classification, especially deep learning models. The prediction is done based on the similarity of the validation image to the training images. The image segmentation block separates the images into two pools: the testing set, and the validation set is shown in Figure 1. The core of the system is a pre-trained machine learning model for image classification, which could be a deep learning model (such as a CNN) or a traditional machine learning model, depending on the task's complexity. The system preprocesses uploaded images to ensure consistency with the training data, involving feeding the images and their corresponding labels to the algorithm and adjusting the model's internal parameters to minimize classification errors. The testing set is used for training the model, while the validation pool is used for testing and predicting.

The proposed object detection system for user-generated data using CNNs focuses on customizing a CNN model tailored to the specific characteristics of user-provided images. The system begins with the collection and preprocessing of user-generated data, involving normalization and data augmentation techniques to enhance model robustness against variations in the data. A custom CNN architecture is designed, incorporating several convolutional and pooling layers to extract hierarchical features, followed by fully connected layers to predict bounding boxes and class probabilities. The training process involves a combined loss function to optimize localization, confidence, and classification accuracy, using user-generated labeled data for effective learning. During inference, the model processes new user-generated images, predicting object locations and classes. Post-processing techniques, including non-maximum suppression (NMS), are applied to eliminate redundant bounding boxes and retain the most confident detections. This system adapts to the specific nature of user-generated data, providing accurate and efficient object detection for personalized applications such as social media content analysis, user-uploaded image tagging, and custom surveillance systems.

In the image acquisition block, the image is sourced either from the internet or from the camera. Image acquisition block represents the raw image dataset you collected for training and testing the classifier. The image segregation block helps in separating the Images into two pools, namely, testing set and the validation set. A portion of the pre-processed data with corresponding class labels is used for training the model. The testing set is used for training the model while the validation pool is used for testing and predicting. The model is trained on the labeled data. This involves feeding images and their labels to the algorithm and adjusting its internal parameters to minimize errors. The prediction is done based on the similarity of the validation image to the training images.



**Figure 1.** Block diagram of image classifier using machine learning.

## LITERATURE SURVEY

Schroder et al. [1] developed a system using Bayesian classifiers to represent high-level land cover labels for pixels using their low-level spectral and textural attributes. They used these classifiers to retrieve images from remote sensing archives by approximating the probabilities of images belonging to different classes using pixel-level probabilities [1].

Koperski et al. [2] developed the VisiMine system, which supports interactive classification and retrieval of remote sensing images by extending content modeling from pixel level to region and scene levels.

Blankenship and Alouani [3] described the classification of images using artificial neural networks (ANN). The system captured and scanned images using Matlab Neural Networks Toolbox and supplied images to the computer via Vitec 2000 Image Card. Features were obtained by color inverting and counting the number of pixels with a value of one. Classification was done with perceptron neural networks, though the paper lacks a discussion on accuracy [3].

Shaw [4] discussed the early work on syntactical descriptions of images, including the picture description language based on operators that represent concatenations between elementary picture components like line segments in line drawings.

Jain and Niblack [5] highlighted advances in databases and computer vision research that resulted in methods for image archiving, retrieval, and image database (IDB) design work. They emphasized the need for increased communication between the vision and database communities to address these issues [5].

Deng et al. [6] introduced ImageNet, a comprehensive database of more than 1.2 million categorized natural images of over 1,000 classes, significantly enhancing computer vision research.

Girshick et al. [7] demonstrated that CNN models trained on the ImageNet database serve as the backbone for improving many object detection and image segmentation problems using other datasets.

Roth et al. [8] used a decompositional 2.5D view resampling and an aggregation of random view classification scores to eliminate the “curse-of-dimensionality” issue, thereby acquiring a sufficient number of training image samples.

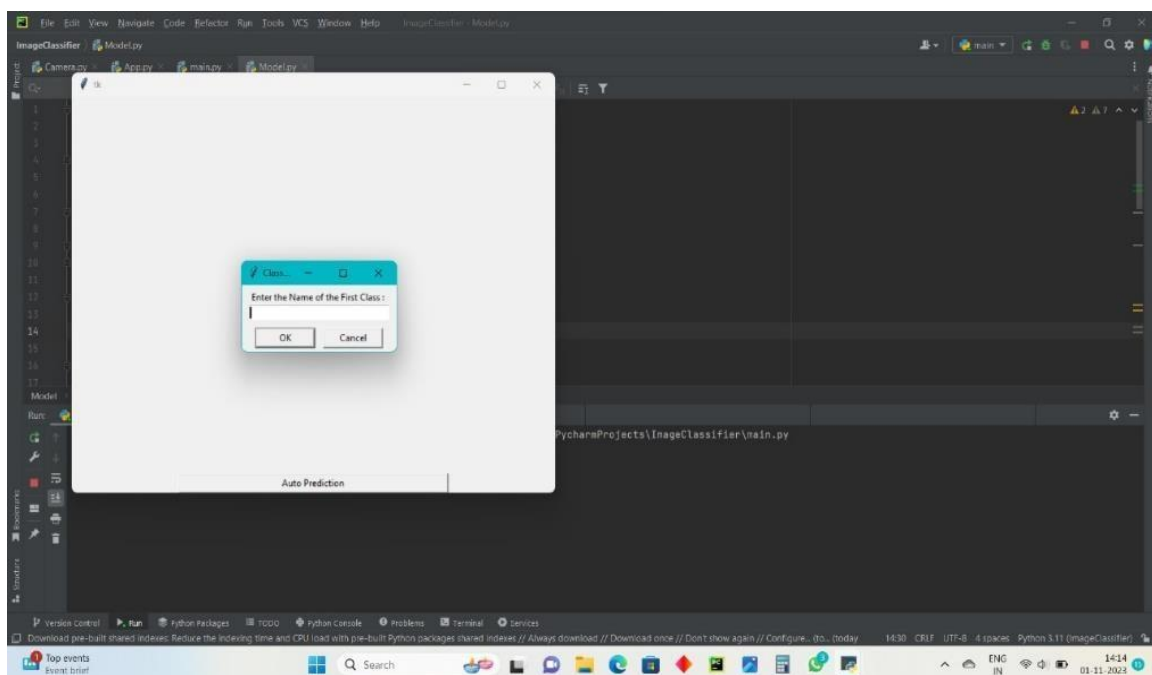
Wu [9] proposed histogram intersection kernel (HIK) support vector machine (SVM) for image classification, introducing intersection coordinate descent (ICD), a deterministic and scalable HIK SVM solver, which is much faster and has similar accuracies to general-purpose SVM solvers and other fast HIK SVM training methods [9].

Krizhevsky et al. [10] introduced the AlexNet architecture, which significantly advanced the field of deep learning for image classification by winning the ImageNet Large Scale Visual Recognition Challenge (ILSVRC).

## RESULTS

This file is crucial for training, evaluating, and using the model to make predictions on new data. Here is a description of what functionalities a model.py file might include, addressing prompts related to class names and model configuration. This includes specifying layers, activation functions, and parameters for building the neural network or other machine learning model.

The results section showcases the graphical user interface (GUI) and outputs of the project. Figure 2 illustrates the GUI, which is essential for training, evaluating, and using the model to make predictions on new data. This file includes functionalities related to class names and model configuration, specifying layers, activation functions, and parameters for building the neural network or other machine learning model.



**Figure 2.** Graphical user interface (GUI).

Figure 3(a) shows the acquisition of images through a camera interface, with the captured images subsequently stored for prediction purposes. This workflow involves gathering a dataset of images to be used for analyzing and classifying various objects or scenes. The stored images serve as input data for a prediction system designed to generate accurate classifications based on machine learning algorithms.

The result (Figure 3(b)) refers to the outcome of the image classification process. It includes the predicted labels or categories assigned to the input images based on the model's analysis.

The result (Figure 3(c)) may indicate the accuracy of the predictions and the model's confidence level in those predictions. Higher confidence values typically suggest more reliable classifications.

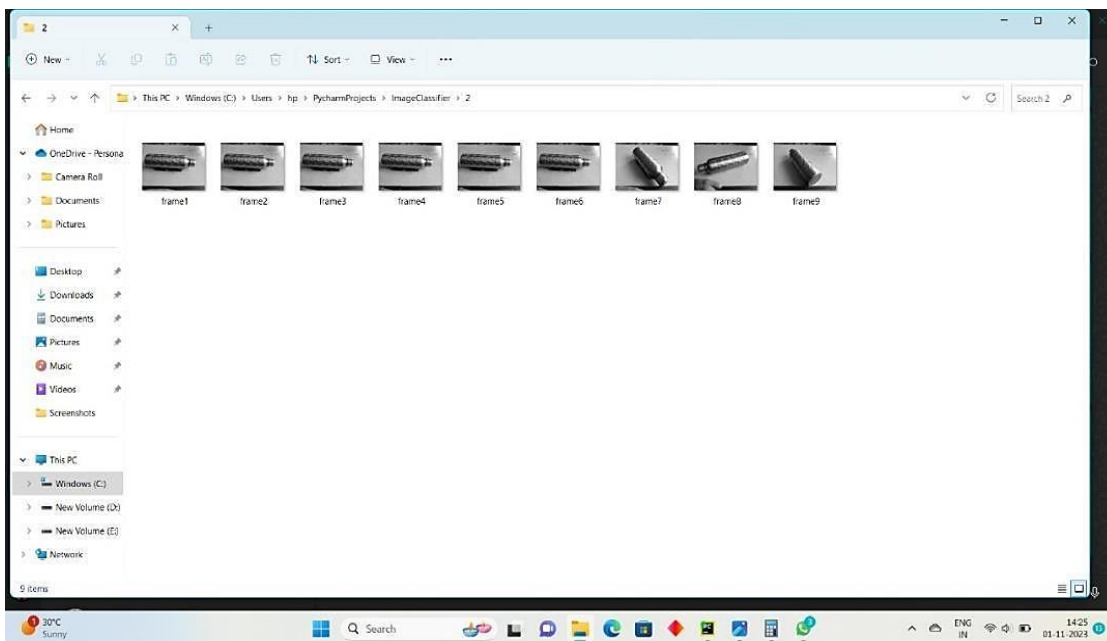


Figure 3. (a) Captured images of the project.

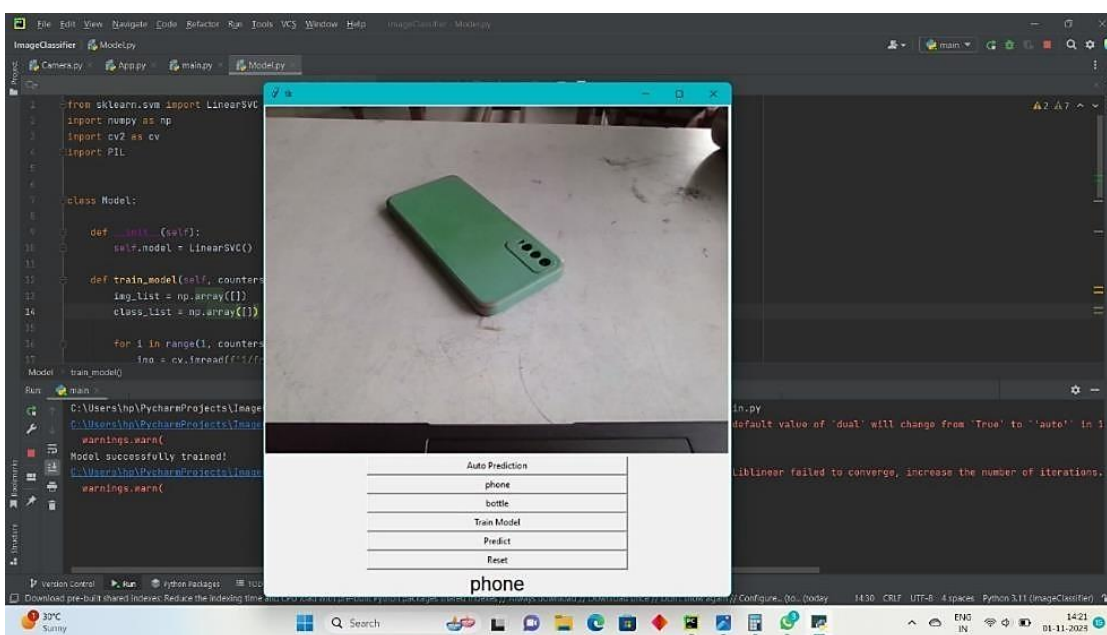
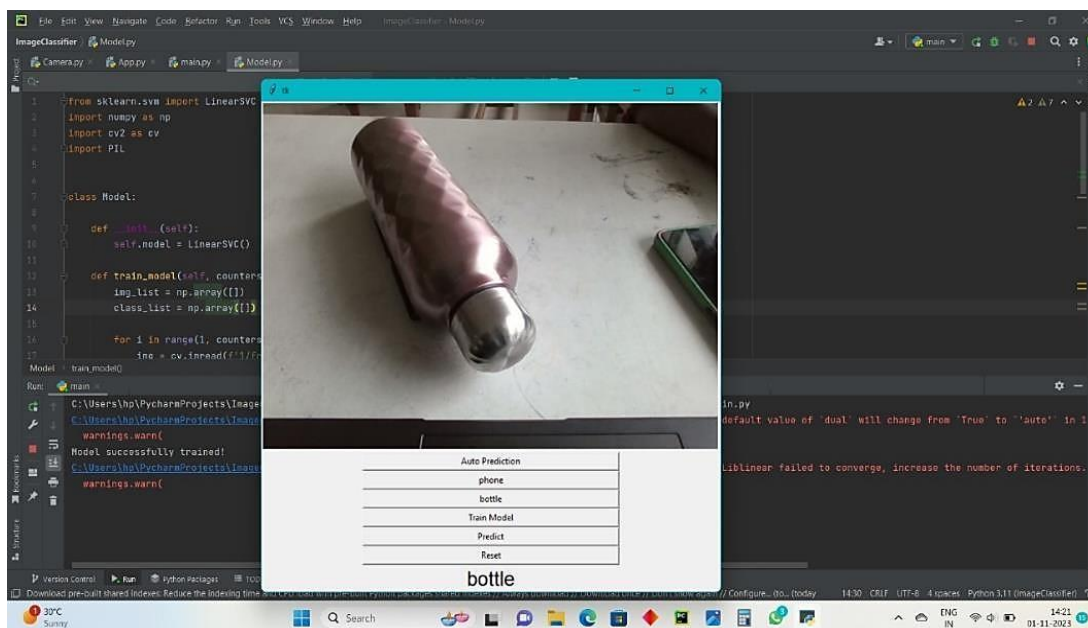


Figure 3. (b) Model's analysis of project-1.



**Figure 3.** (c) Model's analysis of project-2.

## CONCLUSION

With the help of this system, we would be able to classify the images uploaded to the classifier system into one of the classes defined in the system. With this system, it becomes easy and reduces time required for image segregation by use of machine learning. These results demonstrate the power and potential of image classification using machine learning.

## Acknowledgments

We wish to convey our sincere thanks and gratitude to the management of our college and our respected principal Dr. A. V. Deshpande for providing all necessary facilities. We express our heartfelt thanks to our Head of Department, Dr. S.K. Jagtap, Department of Electronics and Telecommunication, for providing constructive suggestions and encouragement during the project. Our sincere thanks to our project guide Mr. S. K. Shinde and mega project coordinators Mr. P. S. Kokare and Ms. M. M. Sonkhaskar for their guidance and support. We extend our gratitude to all the teaching and non-teaching faculty of our department for their cooperation and suggestions during the project work.

## REFERENCES

1. Schroder M, Rehauer H, Siedel K, Datcu M. Interactive learning and probabilistic retrieval in remote sensing image archives. *IEEE Trans Geosci Remote Sensing*. 2000; 38 (5): 2288–2298.
2. Koperski K, Marchisio G, Aksoy S, Tusk C. VisiMine: interactive mining in image databases. In: *IEEE International Geoscience and Remote Sensing Symposium*, Toronto, Ontario, Canada, June 24–28, 2002. Vol. 3, pp. 1810–1812.
3. Blankenship B, Alouani AT. Design and Implementation of a vision system using a neural network classifier. In: *Proceedings The Twenty-Ninth Southeastern Symposium on System Theory*, Cookeville, TN, USA, March 9–11, 1997. pp. 29–33.
4. Shaw AC. Parsing of graph-representable pictures. *J ACM*. 1970; 17 (3): 453–481.
5. Jain R, Niblack W. In: *Proceedings of the NSF Workshop on Visual Information Management*, Redwood, CA, USA, February 24–25, 1992.
6. Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L. ImageNet: a large-scale hierarchical image database. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Miami, FL, USA, June 20–25, 2009. pp. 248–255.
7. Girshick R, Donahue J, Darrell T, Malik J. Region-based convolutional networks for accurate object detection and semantic segmentation. *IEEE Trans Pattern Anal Mach Intell*. 2015; 38 (1): 142–158.

8. Roth H, Lu L, Liu J, Yao J, Seff A, Cherry KM, Turkbey E, Summers RM. Improving computer-aided detection using convolutional neural networks and random view aggregation. *IEEE Trans Med Imaging*. 2016; 35 (5): 1170–1181.
9. Wu J. Efficient HIK SVM learning for image classification. *IEEE Trans Image Process*. 2012; 21 (10): 4442–4453. doi: 10.1109/tip.2012.2207392.
10. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. *Commun ACM*, 2017; 60 (6): 84–90.