MINISTRY OF EDUCATION AND RESEARCH OF THE REPUBLIC OF MOLDOVA
Technical University of Moldova
Faculty of Computers, Informatics, and Microelectronics
Department of Software Engineering and Automation

Approved for defense
Department head:
Ion FIODOROV, phd, associate professor
____
“___” ___________ 2025

ANALYSIS AND RESEARCH OF COMPUTER VISION METHODS FOR OBJECT RECOGNITION IN IMAGES

Master's project

Student:	**Cernev Evgheni, IS-221M**
Coordinator:	**Catruc Mariana,** **university lecturer**
Consultant:	**Cojocaru Svetlana,** **university assistant**

Chișinău, 2025

ABSTRACT

Această lucrare descrie o aplicație mobilă pentru barmani care utilizează tehnici de viziune computerizată și rețele neuronale pentru a identifica ingredientele băuturilor. Această cercetare urmărește să dezvolte un sistem care poate prezenta o selecție de imagini ale ingredientelor, să le recunoască automat și să producă recomandări de cocktailuri corespunzătoare care pot fi făcute cu aceste ingrediente.

Aceasta creează o aplicație independentă de platformă folosind react native și utilizează TensorFlow Lite pentru prelucrarea locală a imaginilor. Aceasta utilizează arhitectura modelului YOLOv8 și mai multe seturi de date specializate pentru a antrena modelul în vederea preciziei în recunoașterea obiectelor.

Acesta descrie etapele de pregătire a datelor și formarea și reglarea modelului, atât de bine a explicat interfața și implementarea serverului aici Acesta include caracteristici pentru salvarea și filtrarea rețetelor, funcționarea offline și integrarea în cloud.

Rezultatele testelor indică faptul că sistemul are o acuratețe generală de 92%, cu o precizie și o completitudine de 90% și, respectiv, 88% în timp ce este în joc în medii diverse, cu rezultate consecvente și repetabile. Pe dispozitivele mobile mid-range, modelul optimizat oferă un timp de procesare a imaginilor de <1 s per imagine.

Ca parte a acestei lucrări, a fost dezvoltată o aplicație prototip pentru a oferi o interfață pentru ca un utilizator să încarce o imagine și să caute o rețetă. Sistemul ar putea fi dezvoltat în continuare și introdus în alte domenii, cum ar fi medicina și logistica.

Aceasta este o aplicație mobilă care utilizează viziunea computerizată și rețele neuronale pentru a identifica rețete de cocktailuri din fotografii făcute de utilizatorul aplicației. Cuvinte-cheie: aplicație mobilă, computer vision, rețele neuronale, TensorFlow Lite, React Native, recunoașterea obiectelor, rețete de cocktailuri.

ABSTRACT

This work describes a mobile application for bartenders that utilizes Computer vision techniques and neural networks to identify drink ingredients. This research seeks to develop a system that can present a selection of ingredient images, automatically recognize them, and output corresponding cocktail recommendations that can be made with these ingredients.

It creates platform independent application using react native and use TensorFlow Lite for local image processing. It utilizes YOLOv8 model architecture and multiple specialized datasets to train the model for accuracy in recognizing objects.

It describes the steps of data preparation and the model training and tuning, so well explained the interface and server side implementation here It includes features for saving and filtering recipes, offline operation and cloud integration.

Test results indicate that the system has an overall accuracy of 92% with precision and completeness at 90% and 88% respectively while in play in diverse environments with consistent and repeatable results. On mid-range mobile devices, the optimized model offers an image processing time of <1 s per image.

As part of this work a prototype application has been developed to provide an interface for a user to upload an image and to search for a recipe. The system could be further developed and introduced to other fields, like medicine and logistics.

This is a mobile application that uses computer vision and neural networks to identify cocktail recipes from pictures taken by the user of the app. Keywords: mobile application, computer vision, neural networks, TensorFlow Lite, React Native, object recognition, cocktail recipes.

TABLE OF CONTENTS

LIST OF TERMS

Accuracy - A metric reflecting the proportion of correctly predicted objects out of the total predictions.

Precision - A metric evaluating the proportion of true positive predictions among all predictions made.

Recall - A metric showing the proportion of actual objects correctly identified by the model.

IoU (Intersection over Union) - A metric for assessing object localization quality by measuring the overlap between the predicted and ground truth bounding boxes.

Quantization - A model optimization method reducing weight sizes to 8-bit format.

TensorFlow - A platform for building and training machine learning models.

TensorFlow Lite - A lightweight version of TensorFlow designed for mobile and embedded devices.

Vision Transformers (ViT) - A modern neural network architecture leveraging attention mechanisms.

Dropout - A regularization technique to reduce the risk of overfitting.

Batch Normalization - A method for normalizing input data to accelerate training.

Data Augmentation - A technique for increasing dataset size by modifying images (rotation, brightness adjustments, etc.).

Modular Architecture - A system structure divided into independent modules.

Self-Supervised Learning - A learning approach where models learn from unlabeled data using pretext tasks.

Neural Network - A set of algorithms modeled after the human brain, designed to recognize patterns.

Bounding Box - A rectangle used to define the location of an object in an image.

Inference - The process of making predictions using a trained model.

Regularization - Techniques to improve model generalization and prevent overfitting.

Real-Time Object Detection - The ability to identify objects in a video stream instantaneously.

API Gateway - A centralized access point for routing requests between microservices.

Cloud Computing - The use of cloud platforms like AWS, Google Cloud, or Azure for resource-intensive tasks.

GPU Acceleration - Using Graphics Processing Units to speed up model training and inference.

Edge Computing - Processing data near the source, such as on mobile devices, rather than in centralized servers.

Embedded Systems - Computing systems designed for specific tasks within larger systems (e.g., IoT devices).

Cross-Platform Development - Creating applications compatible with multiple operating systems using shared codebases.

Object Localization - The process of identifying the position of objects in an image.

INTRODUCTION

Over the past decades, modern computer vision technologies evolved into one of the hottest trends in artifical intelligence. These are the technologies that are used a lot, for addressing the tasks that could only be accomplished by Human till then. For instance, computer vision can be used for image and video analysis to identify, track and classify objects by different properties. These advances paved the way for the development of smart systems that can read and understand the visual content to make decisions on it [1].

Computer vision is one of the most comprehensively used fields today. For instance, in medicine, there are image analysis algorithms that helps doctors to diagnose complicated diseases by analyzing X-rays, CT or MRI scans [2]. The accuracy of diagnostics is enhanced with these technologies, which lowers the chances of errors and speeds up the process of diagnosis [3].

For example, in manufacturing, mechanics automate quality control of products using computer vision [4]. Detection of defects at different stages of production with high reliability and accuracy is attainable with systems based on such technologies [5]. So, when it comes to mass production, this is vital because human control is so cryptic.

Computer vision is one of the most important tools in the security field. The Face and object recognition capabilities lets to recognize people in public places and predict threats, providing security of transport, public places and objects of strategic importance [6].

Computer vision in entertainment create more realistic and interactive applications. For instance, augmented reality technologies which include the filtering functions for social network visuals or applications for interior designing are grounded on algorithms for the expression of an image and recognition of an object [7].

Image object recognition is among the fundamental problems in computer vision. It comprises of object detection and image [8]. Deep learning, especially convolutional neural networks (CNNs) has revolutionised object recognition in terms of accuracy and speed [9]. There are some advantages and disadvantages of modern approaches depending on the application area [10].

In this work, we focus primarily on using the currently known and relatively popular object recognition methods and their application to a specific problem, namely identifying drinks in an image and filtering cocktail recipes that can be prepared from these drinks. The project consists of a development of a mobile application that enables users to sign up, view a list of cocktail recipes, search for recipes based on their name, use filters, and submit photos of drinks taken using the camera or choose photos from the gallery. The computer vision model detects drinks from the photo and recommends mixing recipes for the ingredients that were detected.

The goal is to develop a system which is able to correctly recognize drinks in images and provides helpful suggestions to the user. With the above goal in mind, the objectives were as follows:

- review of existing object recognition methods, including classical algorithms (SIFT, HOG) and modern approaches based on deep neural networks (YOLO, SSD, Mask R-CNN) [11][12];
- comparative analysis of algorithms by criteria such as accuracy, speed of operation and robustness to external factors (complex backgrounds, illumination changes) [13];
- experimental testing of the selected methods using beverage images;
- formulating conclusions and recommendations on the applicability of the methods for mobile devices with limited computational resources.

The LabelImg application [14] was used to create the training sample for the project. It was used to create and label annotations of objects (drinks) in images, which provided high quality data for training the YOLOv8 model [15].

The following tools and technologies are utilized as part of the work:

- TensorFlow for model development and training [16];
- OpenCV for image preprocessing [17];
- LabelImg for creating and marking object labels [14];
- Matplotlib and Seaborn to visualize the experimental results [18];
- React Native and Tailwind CSS to create the user interface of a mobile application [19].

The results of the works will be practical and can be used for computer vision integration to mobile applications. An extensive effort focuses on model compression for deployment on mobile devices [20].

Hence, this paper not only conducts a mini-survey of recent object recognition techniques, but also shows their usage to implement an assistant system to process the images and provide recommendations, which can be further developed and researched.

LIST OF SOURCES USED

[1] Goodfellow, I., Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016.

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436-444, 2015.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in Advances in Neural Information Processing Systems, vol. 25, 2012.

[4] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," arXiv preprint arXiv:1804.02767, 2018.

[5] W. Liu et al., "SSD: Single shot multibox detector," in European Conference on Computer Vision, pp. 21-37, 2016.

[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in Advances in Neural Information Processing Systems, vol. 28, 2015.

[7] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700-4708, 2017.

[8] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in International Conference on Machine Learning, pp. 6105-6114, 2019.

[9] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," arXiv preprint arXiv:2010.11929, 2020.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770-778, 2016.

[11] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," International Journal of Computer Vision, vol. 115, no. 3, pp. 211-252, 2015.

[12] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.

[13] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1251-1258, 2017.

[14] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," in European Conference on Computer Vision, pp. 740-755, 2014.

[15] M. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous systems," arXiv preprint arXiv:1603.04467, 2016.

[16] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in Advances in Neural Information Processing Systems, vol. 32, 2019.

[17] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 886-893, 2005.

[18] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, vol. 60, no. 2, pp. 91-110, 2004.

[19] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," Computer Vision and Image Understanding, vol. 110, no. 3, pp. 346-359, 2008.

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.

[21] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in Proceedings of COMPSTAT'2010, pp. 177-186, 2010.

[22] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 8, pp. 1798-1828, 2013.

[23] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in International Conference on Machine Learning, pp. 448-456, 2015.

[24] T.-Y. Lin et al., "Feature pyramid networks for object detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2117-2125, 2017.

[25] P. Viola and M. J. Jones, "Rapid object detection using a boosted cascade of simple features," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. I-I, 2001.

[26] M. Everingham et al., "The Pascal Visual Object Classes (VOC) challenge," International Journal of Computer Vision, vol. 88, no. 2, pp. 303-338, 2010.

[27] J. Deng et al., "ImageNet: A large-scale hierarchical image database," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 248-255, 2009.

[28] R. Szeliski, Computer Vision: Algorithms and Applications. Springer, 2010.

[29] R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision. Cambridge University Press, 2003.

[30] D. A. Forsyth and J. Ponce, Computer Vision: A Modern Approach. Prentice Hall, 2002.

[31] T. Hastie, R. Tibshirani, and J. H. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, 2009.

[32] J. Schmidhuber, "Deep learning in neural networks: An overview," Neural Networks, vol. 61, pp. 85-117, 2015.

[33] K. P. Murphy, Machine Learning: A Probabilistic Perspective. MIT Press, 2012.

[34] F. Chollet, Deep Learning with Python. Manning Publications, 2018.

[35] D. E. King, "Dlib-ml: A machine learning toolkit," Journal of Machine Learning Research, vol. 10, pp. 1755-1758, 2009.

[36] E. Alpaydin, Introduction to Machine Learning. MIT Press, 2020.

[37] C. Szegedy et al., "Going deeper with convolutions," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1-9, 2015.

[38] R. Girshick et al., "Rich feature hierarchies for accurate object detection and semantic segmentation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580-587, 2014.

[39] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in European Conference on Computer Vision, pp. 818-833, 2014.

[40] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," Science, vol. 313, no. 5786, pp. 504-507, 2006.

[41] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in Neural Networks: Tricks of the Trade. Springer, 2012.

[42] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," Journal of Machine Learning Research, vol. 9, pp. 2579-2605, 2008.

[43] M. Abadi et al., "TensorFlow: A system for large-scale machine learning," in Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation, 2016.

[44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.

[45] L. Bottou, "Online learning and stochastic approximations," in On-line Learning in Neural Networks, 1998.

[46] Z. H. Zhou, Machine Learning. Springer, 2021.

[47] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction. MIT Press, 2018.

[48] T. M. Mitchell, Machine Learning. McGraw-Hill, 1997.

[49] C. M. Bishop, Pattern Recognition and Machine Learning. Springer, 2006.

[50] Dlib Team, "Dlib library documentation," Available online: dlib.net, 2020.

[51] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015

[52] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in Proceedings of the IEEE International Conference on Computer Vision, pp. 2980–2988, 2017.