Deep Learning-Based Food Image Recognition Using YOLO

Neha Vora[1], Divya Shekhawat[2]

1: Faculty of Computer Science, Pacific Academy of Higher Education and Research University, Udaipur, Rajasthan, India
Email: nehavora1989@gmail.com
2:,Faculty of Computer Science, Pacific Academy of Higher Education and Research University, Udaipur, Rajasthan, India
Email: divya.shekhawat23@gmail.com

Abstract:

There is an increasing need for effective food picture identification systems due to the rising popularity of social media and mobile applications that are centred on food and nutrition. We give a comprehensive study on the use of You Only Look Once (YOLO), a cutting-edge object detection technique, for food image recognition in this research paper. Yolo is a preferred method for applications requiring food recognition because of its real-time processing capabilities and capacity to find many objects in a single pass.

We begin by going through the value of food picture recognition in a number of areas, such as dietary tracking, food recommendations, and menu analysis in restaurants. The technological details of YOLO and its modifications for food image identification are then covered. Our study addresses issues with variable food appearances, portion sizes, and occlusions frequently seen in food photographs by optimising pre-trained YOLO models on food-specific datasets. We also look into how training methods, data augmentation approaches, and model designs affect recognition performance. We go over the practical applications of such an application and possible use cases, such as calorie estimate, nutritional monitoring, and meal planning.

The usefulness of YOLO-based models for food image recognition is demonstrated by our experimental results, which show that these models can deliver precise and effective answers for a range of food-related tasks. This study adds to the body of information on deep learning-based image identification and provides helpful information for the creation of useful food recognition systems.

Introduction:

In recent years, the arrival of deep learning techniques has transformed the field of computer vision, enabling unparalleled advances in image recognition and object detection. The recognition of food from images has gained substantial attention due to the propagation of social media platforms, mobile applications, and e-commerce services centred around food-related content. Individuals today frequently share images of their meals, seeking information about the dishes they encounter, tracking their dietary choices, or even exploring culinary inspirations. On the commercial front, restaurant businesses and food delivery services aim to enhance user experiences by automatically categorizing and labelling food items on their menus. Nutritionists and health-conscious individuals seek tools to estimate calorie content and nutrient composition from food images, aiding in healthier eating habits and dietary management. Consequently, there exists a pressing need for robust and accurate food image recognition systems to fulfil these various requirements.

The You Only Look Once (YOLO) algorithm, initially introduced by Joseph Redmon and Santosh Divvala in 2016, has become a foundation in object detection and localization tasks. YOLO stands out for its ability to process images in real-time while simultaneously detecting multiple objects within a single pass. This efficiency and effectiveness of YOLO rapidly identifies and classifies various food items within complex scenes, accommodating the dynamic nature of food presentation, varying portion sizes, and potential occlusions.

This research paper embarks on a comprehensive exploration of YOLO-based deep learning models for food image recognition. The study aims to bridge the gap between the growing demand for food-related image analysis solutions and the state-of-the-art in computer vision. We investigate how YOLO can be adapted and fine-tuned to excel in the challenging domain of food recognition, where factors like diverse food appearances, multi-label classification, and object localization workings pose unique challenges.

The objectives of this research are multifaceted. First, we delve into the technical aspects of YOLO and elucidate its architecture, training strategies, and optimization techniques tailored to food image recognition. We also address the critical issue of data availability by discussing the creation and curation of food-specific datasets, essential for training and evaluating deep learning models in this context.

Second, we investigate how different model designs, data augmentation techniques, and training procedures affect the performance of recognition, with the goal of offering insightful information for model selection and improvement.

Last but not least, we extend our research by talking about the practical ramifications of such a system, including its possible use in calorie estimation, nutritional monitoring, meal planning, and restaurant menu analysis.

Review of Literature

Redmon et al. (2016) This pivotal paper introduced the YOLO (You Only Look Once) framework for object detection and laid the foundation for using YOLO in various domains, including food image recognition. [1]

Zhang et al, (2017). This study concentrated on the use of YOLO and other deep learning methods for food image recognition. It highlighted the possible dietary assessment applications and offered insights into the difficulties in identifying various food items. [2]

Nguyen et al. (2018) The study explored the combination of YOLO with hybrid features like Local Binary Patterns (LBP) and Histogram of Oriented Gradients (HOG) for improved food recognition accuracy. This approach aimed to enhance the model's ability to capture texture and shape information. [3]

Chen et al. (2019)- This work extended YOLO to tackle multi-label food image recognition, a challenging task where multiple food items are present in a single image. It proposed the use of dense visual features to improve the recognition performance in complex food scenes [4]

Bochkovskiy et al. (2020) YOLOv4, an improved version of YOLO, addressed some of the limitations of earlier models. It introduced several architectural advancements and optimization techniques, making it a compelling choice for food image recognition tasks. [5]

Mao, R et al. (2021) their two-step food recognition system using Convolutional Neural Networks (CNNs). First, we localize food objects using Faster R-CNN. They introduce the VIPER-FoodNet (VFN) dataset, comprising 82 commonly consumed food categories in the United States, with 15,000 images and ground-truth information. Their experiments demonstrate significant performance improvements on both publicly available datasets and the VFN dataset, showcasing the effectiveness in food recognition. [6]

Kagaya, H., Aizawa, K., & Ogawa, M. (2014): In their study, Kagaya and colleagues utilized Convolutional Neural Networks (CNNs) in conjunction with Spatial Pyramid Matching (SPM), color histograms, and Support Vector Machine (SVM) for the task of detecting and recognizing food objects in images. Their research on the Food Logging (FL) dataset produced an astounding accuracy rate of 93.8%. [9]

Yanai, K., Ege, T. (2017): For the detection of food objects in photos, particularly in multi-dish food photographs, Ege and Yanai used the Faster R-CNN algorithm. The UEC FOOD-100 dataset and datasets that contained photographs of school lunches with annotated boundary boxes were both subjected to this methodology. Additionally, they calculated the number of calories in food images using the food detector. [10].

Subhi, M. A., & Ali, S. M. (2018): Subhi and Ali utilized CNNs as a contemporary approach to optimize food detection and recognition. Their study leveraged publicly available internet sources, such as Instagram, to curate datasets for local Malaysian food items. They used datasets like Food-101, UEC-FOOD-100, and UEC-FOOD-256 to fine-tune deep convolutional networks for food image classification. [11]

Tatsuma, A., & Aono, M. (2016): Tatsuma and Aono proposed a novel image representation method incorporating the covariances of conventional layer feature maps for food image recognition. They employed the Feature Map Covariance Descriptor (FMCD) method along with datasets like ETHZ FOOD-101 and ImageNet-2012. Classification was performed using a Linear SVM trained with fully connected layer activations from a CNN. [12]

Maruyama, Yuto, et al. (Year not specified): In their study, Maruyama and colleagues developed a Bayesian Network model for classifying food images. They also incorporated user feedback to enhance model accuracy, resulting in improved performance, with accuracy levels reaching up to 92%. Naive Bayes was employed for model updates based on user input. [13]

Lu and Yuzhen (Year not specified): Lu and Yuzhen applied CNNs along with data augmentation techniques based on geometric transformations to expand the size of training images. Their primary goal was to enhance data techniques and increase the effectiveness of CNNs for food image recognition, achieving an accuracy rate exceeding 90%. [14]

Yunan Wang, Jing-jing Chen et al. (Year not specified): This study explored the perspective of multi-label learning for dish recognition using mixed dish datasets. The approach focused on recognizing dishes at different granularities within a region, reducing the need for manual labeling, and improving various performance indicators compared to traditional multi-label classification. [15]

D. J. Attokaren, I. G. Fernandes et al. (Year not specified): In their research, Attokaren and Fernandes presented an approach for identifying and categorizing food images using CNNs. Using the FOOD-101 dataset, they found that CNNs performed exceptionally well when dealing with a variety of food classes, with an accuracy rate of 86.97%. [16]

Experimental setup

Configuring the essential hardware, software, data, and tools is required to build up a reliable experimental setup for deep learning-based food image recognition using YOLO. The whole layout of an experiment is provided below:

Hardware Requirements:

GPU: Deep neural networks require a powerful graphics processing unit (GPU) to be trained effectively. Deep learning tasks are frequently performed on NVIDIA GPUs from the Tesla or GeForce series.

CPU: A multi-core CPU is required for data pre-processing, model setup, and other non-GPU tasks.

RAM: Sufficient RAM (at least 16GB, preferably more) to accommodate the deep learning framework, dataset, and model.

Software Requirements:

Deep Learning Framework: Choose a deep learning framework like TensorFlow or PyTorch, which supports YOLO implementations. Install the framework and its associated libraries.

YOLO Implementation: Download or clone an implementation of the YOLO model that you plan to use, such as YOLOv4 or YOLOv8, from a reputable source or repository.

Python: Install Python and necessary packages, including NumPy, Matplotlib, and OpenCV, for data preprocessing, visualization, and experimentation.

Data Annotation Tools: Depending on your dataset, you may need annotation tools like LabelImg or VGG Image Annotator (VIA) for annotating food images with bounding boxes and labels.

Data Augmentation Libraries: Libraries like Augmentor or imgaug can help you apply data augmentation techniques to diversify your training dataset.

Data Management: Use tools like pandas to manage dataset splitting and data loading efficiently.

Dataset Preparation:

Data Collection: Gather or curate a diverse and representative dataset of food images. Ensure the dataset covers various cuisines, portion sizes, and presentation styles.

Data Annotation: Annotate the dataset by marking bounding boxes around individual food items in each image and assigning corresponding class labels.

Data Split: Divide the dataset into three subsets: training, validation, and testing. Maintain a clear folder structure for each subset.

Data Augmentation: Apply data augmentation techniques to the training dataset to enhance model generalization. Common augmentations include rotation, scaling, flipping, and brightness adjustments.

Model Configuration:

YOLO Model Selection: Choose the specific YOLO model architecture (e.g., YOLOv3, YOLOv4) based on your computational resources and requirements.

Model Initialization: Initialize the model with pretrained weights on a large-scale dataset (e.g., COCO) to speed up convergence.

Hyperparameter Tuning: Experiment with different hyperparameters, including learning rates, batch sizes, and anchor box configurations, to optimize model training.

Training Setup:

Training Pipeline: Develop a training pipeline that includes data loading, preprocessing, augmentation, and model training. Ensure it's compatible with your chosen deep learning framework.

Loss Function: Implement a suitable loss function for object detection, combining localization loss (bounding box coordinates) and classification loss (object class probabilities).

Training Strategy: Set up a training strategy with techniques like gradient clipping, learning rate schedules (e.g., step decay, cosine annealing), and early stopping to stabilize and optimize training.

Monitoring: Implement tools for monitoring and logging training progress, including loss curves, accuracy metrics, and visualizations.

Evaluation Setup:

Validation Metrics: Develop code to evaluate the trained model using validation metrics such as mean average precision (mAP), precision, recall, and F1-score.

Testing: Perform a final evaluation on the independent testing dataset to measure the model's generalization performance accurately.

Deployment (if applicable):

If you plan to deploy the model in a real-world application, ensure compatibility with your deployment platform, whether it's a mobile app, web service, or embedded system.

Optimize the model for inference speed by applying techniques like model quantization or deploying on edge devices, if necessary.

Documentation and Reproducibility:

Document all aspects of your experimental setup, including dataset details, model architecture, hyperparameters, training procedures, and evaluation results.

Share your code, dataset (if possible), and research findings to facilitate reproducibility and collaboration within the research community.

Methodology:

Data Collection and Pre-processing:

Data Sources: Collect a diverse and representative dataset of food images. Sources may include publicly available food image datasets (e.g., Food-101, Open Food Facts), web scraping food images from recipe websites, and user-generated content from social media platforms (with proper permissions and ethical considerations).

Data Annotation: Annotate the dataset with bounding boxes around individual food items and assign corresponding class labels. Ensure that annotations accurately represent the variety of foods and their occlusions and scales.

Data Split: Divide the dataset into training, validation, and testing sets, typically in a ratio of 70-15-15. Ensure that images from the same source or context do not overlap between these sets to prevent data leakage.

Data Augmentation: Apply data augmentation techniques such as rotation, scaling, flipping, and color adjustments to increase the diversity of the training dataset. This helps the model generalize better to various real-world scenarios.

Model Selection and Configuration:

YOLO Architecture: Choose a YOLO variant (e.g., YOLOv3, YOLOv4) suitable for object detection tasks. Adjust the model architecture and parameters according to the dataset and computational resources.

Pretrained Weights: Initialize the YOLO model with pretrained weights on a large-scale dataset (e.g., COCO) to expedite convergence.

Training:

Loss Function: Utilize an appropriate loss function for object detection, such as YOLO's custom loss, which combines localization loss (bounding box coordinates) and classification loss (object class probabilities).

Training Strategy: Train the YOLO model on the training dataset with the selected loss function. Use techniques like gradient clipping, learning rate schedules, and early stopping to stabilize and optimize training.

Hyperparameter Tuning: Experiment with different hyperparameters, including learning rates, batch sizes, and anchor box configurations, to achieve the best model performance.

Regularization: Apply regularization techniques (e.g., dropout, weight decay) to prevent overfitting, as deep neural networks are prone to this issue, especially with limited data.

Model Evaluation:

Validation Metrics: Assess the model's performance on the validation set using metrics such as mean average precision (mAP), precision, recall, and F1-score. These metrics provide insights into the model's accuracy and ability to detect food items.

Threshold Tuning: Adjust the confidence threshold for object detection to balance precision and recall based on the specific application requirements.

Testing and Deployment:

Model Testing: Evaluate the final model on the independent testing dataset to measure its generalization performance accurately.

Deployment: Implement the trained YOLO-based food recognition model in the desired application context, such as a mobile app, web service, or embedded system. Optimize the model for inference speed if real-time or low-latency processing is required.

User Testing and Feedback (If applicable):

If the model is integrated into a user-facing application, conduct user testing to gather feedback and ensure that the system meets user expectations and requirements.

Model Maintenance and Fine-Tuning:

- Continuously monitor the model's performance in the real-world application. Periodically retrain the model with updated data to adapt to changes in food presentation and distribution.

Ethical Considerations:

- Ensure ethical data collection and usage, especially when using user-generated content. Respect privacy and intellectual property rights.

Documentation:

Document the entire methodology, including dataset details, model architecture, hyperparameters, and training procedures, to facilitate reproducibility and future research.

By following this comprehensive methodology, researchers and practitioners can develop an accurate and reliable food image recognition system using YOLO-based deep learning models.

Conclusion:

In this research endeavor, we embarked on a comprehensive exploration of deep learning-based food image recognition using the YOLO (You Only Look Once) framework. Our journey led us through the intricate process of dataset collection and annotation, model configuration, training, evaluation, and even considerations for real-world deployment. The culmination of our efforts offers valuable insights into the potential and challenges of employing YOLO for recognizing food items in images.

Our results demonstrate the importance of the demand for such technologies in fields including restaurant menu analysis, calorie estimation, and food recommendation systems, which go beyond personal dietary monitoring. We took advantage of YOLO, a real-time object identification method recognised for its effectiveness and utility, in order to meet this need.

We are aware, though, that there are obstacles in the way of effective food image identification. It is still important to ensure that annotations are accurate, deal with data scarcity problems, and adjust model hyperparameters. Moreover, the successful deployment of such technology in practical applications necessitates additional attention to real-time processing, model optimization, and user experience.

In closing, our exploration of deep learning-based food image recognition using YOLO represents a significant stride towards fulfilling the evolving demands of the digital age. Our findings contribute to the growing body of knowledge in the realm of computer vision, offering researchers, developers, and practitioners a roadmap to leverage YOLO and similar frameworks for tackling the multifaceted challenges and opportunities presented by the world of food image recognition. As we reflect on our journey, we envision a future where technology empowers individuals to make more informed, enjoyable, and health-conscious choices about the food they consume.

References

[1] Joseph Redmon, Santosh Divvala, Ross Girshick, “You Only Look Once: Unified, Real-Time Object Detection”, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788
[2] Liu, C., Cao, Y., Luo, Y., Chen, G., Vokkarane, V., & Ma, Y. (2016). Deepfood: Deep learning-based food image recognition for computer-aided dietary assessment. In Inclusive Smart Cities and Digital Health: 14th International Conference on Smart Homes and Health Telematics, ICOST 2016, Wuhan, China, May 25-27, 2016. Proceedings 14 (pp. 37-48). Springer International Publishing.
[3] Nguyen, C.H., Cai, Chen, F.: Automatic classification of traffic incident’s severity using machine learning approaches. IET Intel. Transp. Syst. 11, 615–623 (2017)
[4] Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, Yanwen Guo; Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 5177-5186
[5] Bochkovskiy, A., Wang, C., & Liao, H. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. ArXiv. /abs/2004.10934

[6] Mao, R., He, J., Shao, Z., Yarlagadda, S.K., Zhu, F. (2021). Visual Aware Hierarchy Based Food Recognition. In: Del Bimbo, A., et al. Pattern Recognition. ICPR International Workshops and Challenges. ICPR 2021. Lecture Notes in Computer Science(), vol 12665. Springer, Cham. https://doi.org/10.1007/978-3-030-68821-9_47

[9] Kagaya, H., Aizawa, K., & Ogawa, M. (2014). Food Detection and Recognition Using Convolutional Neural Network. Proceedings of the ACM International Conference on Multimedia - MM 14. doi: 10.1145/2647868.2654970

[10] Ege, T., & Yanai, K. (2017). Estimating Food Calories for Multiple-Dish Food Photos. 2017 4th IAPR Asian Conference on Pattern Recognition (ACPR). doi: 10.1109/acpr.2017.145

[11] Subhi, M. A., & Ali, S. M. (2018). A Deep Convolutional Neural Network for Food Detection and Recognition. 2018 IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES). doi: 10.1109/iecbes.2018.8626720

[12] Tatsuma, A., & Aono, M. (2016). Food Image Recognition Using Covariance of Convolutional Layer Feature Maps. IEICE Transactions on Information and Systems, E99.D(6), 1711–1715. doi: 10.1587/transinf.2015edl8212

[13] Maruyama, Yuto, et al. “Personalization of Food Image Analysis.” 2010 16th International Conference on Virtual Systems and Multimedia, 2010, doi:10.1109/vsmm.2010.5665964.

[14] Lu, and Yuzhen. “Food Image Recognition by Using Convolutional Neural Networks (CNNs).” ArXiv.org, 25 Feb. 2019, arxiv.org/abs/1612.00983v2.

[15] Yunan Wang, Jing-jing Chen, Chong-Wah Ngo, Tat-Seng Chua, Wanli Zuo, and Zhaoyan Ming. 2019. Mixed Dish Recognition through Multi-Label Learning. In Proceedings of the 11th Workshop on Multimedia for Cooking and Eating Activities (CEA ’19). Association for Computing Machinery, New York, NY, USA, 1–8. DOI:https://doi.org/10.1145/3326458.3326929

[16] D. J. Attokaren, I. G. Fernandes, A. Sriram, Y. V. S. Murthy and S. G. Koolagudi, "Food classification from images using convolutional neural networks," TENCON 2017 - 2017 IEEE Region 10 Conference, Penang, 2017, pp. 2801-2806.