Vision Based Intelligent Recipe Recommendation System

Tushar Jadhav
Electronics & Telecommunication Department
Vishwakarma Institute of Information Technology
Pune, India
tushar.jadhav@viit.ac.in

Ashwini Navghane
Electronics & Telecommunication Department
Vishwakarma Institute of Information Technology
Pune, India
ashwini.navghane@viit.ac.in

Kunal Avghade
Electronics & Telecommunication Department
Vishwakarma Institute of Information Technology
Pune, India
kunal.22020031@viit.ac.in

Vrushabh Khatik
Electronics & Telecommunication Department
Vishwakarma Institute of Information Technology
Pune, India
vrushabh.22020234@viit.ac.in

Mahavirsingh Rajpurohit
Electronics & Telecommunication Department
Vishwakarma Institute of Information Technology
Pune, India
mahavirsingh.21911186@viit.ac.in

Ruturaj Kalshetti
Electronics & Telecommunication Department
Vishwakarma Institute of Information Technology
Pune, India
ruturaj.21910760@viit.ac.in.in

Abstract—This paper presents the Vision Based Intelligent Recipe Recommendation System (VBIRRS). People who stay remotely away from their homes miss the food at home. Students, Athletes, Professionals etc face the same problem. Although they might know how to cook they may not know which recipes can be made out of available vegetables. VBIRRS provides the solution for this problem. The system is not only intelligent enough to recommend the recipes possible from the available vegetables also remembers the preferences/choices of the user and recommends the recipes smartly. The system uses object detection algorithm YOLO and recommends the Indian recipes. We have achieved the classification accuracy more than 81.6%. To reduce the response time, the rewriting of the bounding box on image after detection is removed which gives a faster response during integration phase. Ajax makes faster reloading of dynamic data to help reducing the response time further**.**

Keywords—Recipe recommendation, YOLO, Confusion Matrix, Ajax

I. Introduction

The introductory requirements of humans are food, water, air, and sanctum. Food plays a major part in our reality. The basic needs of humans are food, water, air, and shelter. Food plays a major role in our existence. Hence, one who knows cooking has an advantage over others. With advancements in technology, people are investing more in health and fitness, but to achieve this goal, one has to eat food as medicine. With technology, one can order food at home.

Although this is a boon to us, a lot of fraud like adulteration, food packaging, unhygienic cooking places, etc. makes the food unhealthy to eat. Eating unhealthy, unhygienic food is similar to providing shelter for diseases. To counter the problem, one can enjoy the food at home, prepared right in your kitchen. Although chefs who know the recipes can't be brought to homes, one can get the detailed recipe procedures with a blend of artificial intelligence. With this, one can enjoy healthy home-cooked food. People who stay remotely away from their homes miss the food at home. Students, Athletes, Professionals etc face the same problem. Some of them might know how to cook. With these things in mind there's a need to create a system to counter all the above problems.

A good system is one which can solve all the above problems and is portable. The user has to take less effort in handling the system and be focused towards the cooking. So the solution is to make the system intelligent using Artificial Intelligence, and to make it portable a website would be a good solution. Hence Vision Based Intelligent Recipe Recommendation System is one in all solutions for our problem. The user has to just spread out the vegetables on a plain area, click an image and upload it to the system, rest of the work is done by the system to provide results.

The system helps people prepare the dish within a few clicks. One can choose from a variety of dishes that have been recommended from their uploaded image of ingredients. The system provides detailed instructions for cooking the dish. The only thing that is important is to dedicate oneself to cooking. Although this would be a little bit tough for beginners, once the user gets a grip, he/she can enjoy the food at home. For people with cooking hobbies, this would be a walk in the park.

II. Literature Survey

[1] Keiji Yanai, Takuma Maruyama and Yoshiyuki Kawano propose a cooking recipe recommendation system that runs on consumers' smartphones. The system detects food ingredients and based on the detected ingredients recommends the cooking recipes. The system was able to recognize 30 kinds of ingredients in 0.15 seconds with an 83.93 recognition rate within the top 6 candidates. For recognition, they built bag-of features by extracting SURF and color histogram from multiple images and used linear kernel SVM at one-vs-all as a classifier.

[2] Xinyu Hu, Yuheng Liu, Yu Xing gave a solution for Food Ingredient Detection for Recipe Recommendation

Systems. They used Mask R-CNN to segment and detect ingredients. They experimented MS-COCO with Decaying Learning rate, L1 and L2 regularization and augmentation on classes of 3 and 6 by a model of 5+ layers and got minimum accuracy of 81.57% and maximum loss of 0.18. ResNet101 was the best model from experimentation. And given results where eggplant and onion were successfully detected and segmented. It resulted in an AP value of 91.60% .

[3] Md. Shafaat Jamil Rokon et al. proposed CNN-based Resnet50 to recognize food ingredients and a recommendation algorithm to suggest cooking recipes. They used the food ingredients dataset of Kaggle and also created a custom dataset of 32 categories. They also experimented detection using VGG16 and MobileNetV2 but ResNet50 generated better testing accuracy of 94%. They selected 19 cooking recipes against 32 food ingredients for recipe recommendation and designed a 2D matrix for the recipe recommendation algorithm. Also Conducted a linear search in the database.

[4] Tossawat Mokdara, Priyakorn Pusawiro, Jaturon Harnsomburana proposed a Recommendation System using Deep Neural Network. They used Thai Food as their test domain. This model uses the user's previous data and analyzes the favorite ingredients of the user using a Deep Neural Network model.Based on the previous data of the user the model predicts the next dishes using a temporal prediction model. The hit ratio is used to calculate the satisfaction of the user. The model can predict with precision up to 90% and the accuracy of the hit ratio is up to 89%.

Sachin C, N Manasa, Vicky Sharma, Nippun Kumaar A. A. proposed Vegetable Classification using You Look Only Once [YOLO] algorithm. They used Tensorflow, and Darkflow for the YOLO algorithm. To train the model images are firstly preprocessed before training by drawing a bounding box around the vegetable manually using OpenCV. The network is trained using the YOLO algorithm.They got an accuracy of 61.6%[5]. Tausif Diwan et al. presented a comprehensive review on single stage object detectors such as YOLOs, performance statistics, and their architecture advancement. They compared detectors in various aspects and found YOLO was faster compared to Faster-RCNN[17].

ZhengXian Li et al. proposed a scalable recipe recommendation system for mobile application. They have designed a hybrid recommendation algorithm, combining content-based and collaborative filtering to improve recommendation system performance. They used spark cluster to process massive data from mobile apps.They have created Android mobile applications as clients where users can interact. Image segmentation is an essential part of deep learning algorithms mainly of deep visual understanding systems. Segmentation is the process of sectioning images into parts where each part resembles an object [6]. Lipi Shah et al. proposed a recommendation system using hybrid approaches which consist of content as well as collaborative filtering, by adding more heterogeneity to reduce RMSE than conventional recommendation systems [12,13].

[7] In this literature they have surveyed image segmentation algorithms based on deep learning models and compared performances of different models. On the other hand Lei Zhou et al. proposed the applications of deep learning in the food domain [8]. One of the biggest advantages of Deep learning technology is Feature learning.

Suyash Maheshwari and Manas Chourey have proposed a recommendation System using machine learning models such as vector- space model and Word2vec model which find top ingredient pairs from various cuisines. Also, they Suggest alternate ingredients [9]. To collect ingredients and recipes data they have used Web scraping technique [9, 10].

Yu-Ru Lin et al. proposed recipe recommendations using ingredient networks. They use ingredients and the relationships encoded between them in ingredient networks to predict recipe ratings, and compare them against features encoding nutrition information [11]. In order to understand users' recipe preferences, they crawled 1,976,920 reviews which include reviewers' ratings, review text, and the number of users who voted the review as useful Jian Chou et al. proposed a solution container management greedy algorithm, which minimizes the probability of generating fragments which saves 10% memory when django templates render [15].

[18] Sonam Khedkar, Swapnil Thube in a research stated Real-time databases help in faster storage and retrieval of data. These databases being lightweight, implements the NoSql engines with the help of BSON which are binary mappings over the data types. The mappings are updated and continuously synchronized with each associated client. The frequent retrievals are cached in order to prevent searching the mappings over the entire database.

Table 1 Comparison of techniques discussed in literature.

Refs.	Works	Algorithm	Datasets	Performance
[1]	A Cooking Recipe Recommendation System with Visual Recognition of Food Ingredients	support vector machine (SVM)	Custom Made Dataset	83.93%
[2]	Food Ingredient Detection For Recipe Recommendation Systems	ResNet101	MS-COCO	81.57%
[3]	Food recipe recommendation based on ingredient detection using deep learning	CNN-based Resnet50	Custom Made Dataset	94%
[5]	Vegetable Classification Using You Only Look Once Algorithm	YOLO	Custom Made Dataset	61.6%

There are various techniques for object detection and classification. Each technique has its merits and defects, which include processing time, storage space and accuracy.

This paper is organized in three sections named as methodology, result and conclusion. Methodology includes the description of used techniques and operational workflow. The result section describes the performance evaluation of the algorithm and achieved outputs.

III. Methodology

The System proposed to provide a solution to recipes recommendation based on detection of ingredients. The system is divided into client server architecture.

The client side user interface or web interface consists of the methods to capture image from users inbuilt camera and sends the request to server to process and detect ingredients from captured image. Based on the response from the server the list of recipes listed on the client web view. Once selecting the recipe from the list the details are fetched from the database.

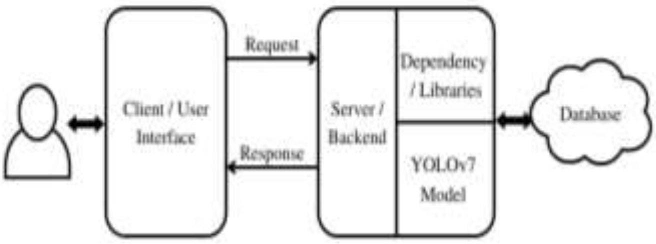

Fig. 1 Conceptual diagram of system

The server processes the request from users and provides a response. The trained Yolov7 network is integrated into the server in order to detect the ingredient. The firebase database consists of NoSql data about recipes, keys and users history. Based on request from the corresponding response is provided

A. *Dataset Construction*

The dataset of vegetable images is used to train the Yolov7 model. The dataset consists of the 15 classes of vegetables and a total 527 images with a total 3253 instances. The dataset processing consists of applied to increase and enhance the features of classes. Each image is annotated and mapped to corresponding class instances.

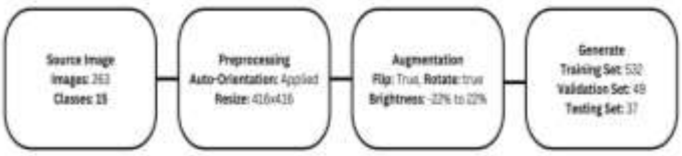

Fig. 2 Dataset Building Process to increase number of Samples

Table 2 Instances of classes

No.	Class Name	Train	Test	Valid	Total
1	Ginger	87	3	10	100
2	Cabbage	96	7	12	115
3	Sponge Gourd	75	6	9	90
4	Tomato	397	24	34	455
5	Onion	300	18	31	349
6	Bitter Melon	187	16	17	220
7	Garlic	261	22	22	305
8	Lady Finger	186	13	20	219
9	Potato	312	19	26	357
10	Cauliflower	120	4	11	135
11	Calabash	93	6	9	108
12	Capsicum	168	9	17	194
13	Brinjal	117	8	13	138
14	Green Chili	294	11	29	334
15	Cucumber	123	3	8	134

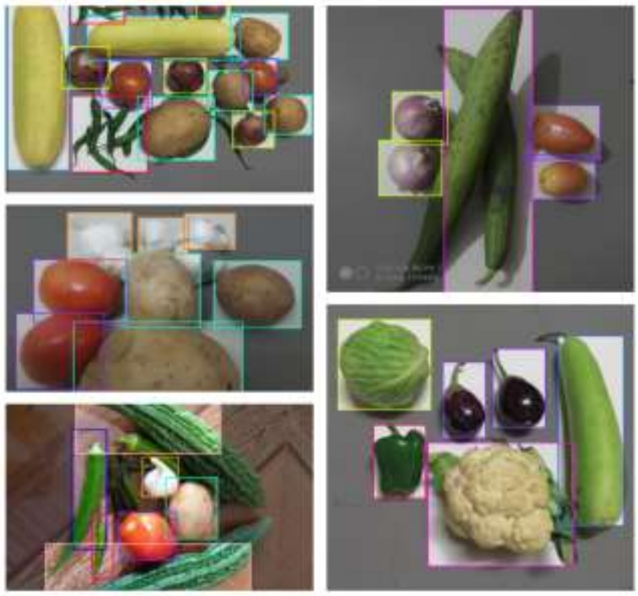

Fig. 4 Dataset Images

Total 263 images are taken through a camera, each image consists of a combination of vegetables and each veggie is labeled to its corresponding class. Image is resized to standard size of 416x416 and augmented by flipping, Rotating and changing brightness levels and the whole dataset splitted at 86% training set, 6% testing set and 8% validation set.

B. *Yolov7 Model*

You Only Look Once (YOLO) v7 is the fastest and most accurate real-time object detection algorithm. Yolo v7 is used as a detection algorithm to detect and classify vegetables from images and provide a list of detected vegetables. It is trained on a custom vegetable dataset with 300 epochs and a Batch size of 16 along with Intersection Over Union (IOU) of 0.2. It detects the ingredients in the frames with the help of convolution neural networks, Residual blocks, bounding box regression, Intersection Over Union (IOU), etc and gives the names of the detected ingredients as output. Yolo v7 is integrated with the processing server as a detection system.

C. *User Interface.*

User interface helps in interacting with the user and visualizes data and response and also sends requests to the server. The Ajax script runs on client side to provide functionality such as accessing user web cam and taking pictures, sending requests, and collecting responses from server. User interface consists of sections for user login, signup, main, contact and about. Each section handles the user by interacting with severe and rendering the response. Main section can be accessed only after the user authenticates, sends the image and based on the response list of the recipe is rendered to the user

D. Processing Server.

The computation and resource providing tasks are handled by the django server. The requests from each view are handled by corresponding handlers and responses sent back to client's. server has Yolo v7 and database along with the NoSql database connectivity. The user Authentication and management handling, and limiting the access to secured sections are limited by server and processed data sent into JSON data format. Once the user sends the image to the server the detection of ingredients is performed to build a list of veggies, Based on the permutation combinations of veggies list of recipes names fetched from NoSql database. The list of recipes sorted as per users previous history. To perform sorting tasks the type of previously selected recipes are used. The list of recipes and detected vegetables are sent to the user as a form of response. When a get recipe request is sent the detailed recipe description is fetched and the user is identified and users history updated based on the current recipes type.

E. NoSql Database.

The Cloud based NoSql Database is capable of handling and storing large quantities of textual data with high availability and responsive nature. The data are arranged in the key value pair and hierarchical rather than stored in a columned table. The access to the database is secured with authentication and query fetched through api.

Recipes	Food	Users
|---Ingredients	|---Recipe Name	|---Username
|---Type of Recipe	|---Info	|---Password
	|---Procedure	|---History
	|---Views	
A. Recipe Table	B. Food Information Table	C. Users Table

Fig. 4 NoSql database tables description.

The recipe name table consists of a combination of veggies as key and list of recipe names as value. The user's history table consists of a username as key and list of types of recipes, recently selected recipe's types are ordered first and previous data are shifted after it on the list, the recipe table uses recipe name as the key and details as value.

IV. Results

In the initial phase of the development, we tested the model on a dataset by piling up a few vegetables on the kitchen counter, and on training, we found that the accuracy of the model averaged 78.3%. Achieving good accuracy and precision was the foundational goal of the entire project, as the model has to identify and classify all the vegetables (the base ingredients). A deep study of the work led us to the conclusion that removing the background noise in the training data would improve the accuracy of the model. After factually implementing the idea, the accuracy was boosted to an average of 81.6%. By combining more vegetable classes, it provided a wide range of ingredients that could be found in a recipe. With a variety of ingredients to identify, the major factor is to avoid false detections on any surface; hence, the dataset was tuned for training with the least difference between the background and instances and with the largest difference between the background and instances. This data, when combined, helped to reduce the major flaws of false detections. Increasing the instances of less accurate classes as well as individualizing the most common ingredients made the model accurate with an astounding average of 88.2%.

Table 3 Performance evaluation.

No.	Class Name	Precision	Recall	Accuracy
1	Bitter melon	0.857	0.765	76%
2	Brinjal	0.84	0.811	77%
3	Cabbage	0.96	1	100%
4	Calabash	0.866	0.889	89%
5	Capsicum	1	0.859	82%
6	Cauliflower	1	0.965	91%
7	Garlic	0.836	0.965	95%
8	Ginger	0.956	0.9	90%
9	Green Chili	0.636	0.724	70%
10	Lady finger	0.636	0.676	79%
11	Onion	1	0.996	100%
12	Potato	1	1	100%
13	Sponge Gourd	0.783	0.806	80%
14	Tomato	0.959	0.971	94%
15	Cucumber	0.959	1	100%

As shown in Table 3. Each class was evaluated based on parameters such as precision, recall, and accuracy. We observed that the vegetables with unique shapes and colors got higher accuracy, while those with small sizes and multiple counts got less accuracy. An increased dataset size from the first trial leads to improved performance on the second trial.

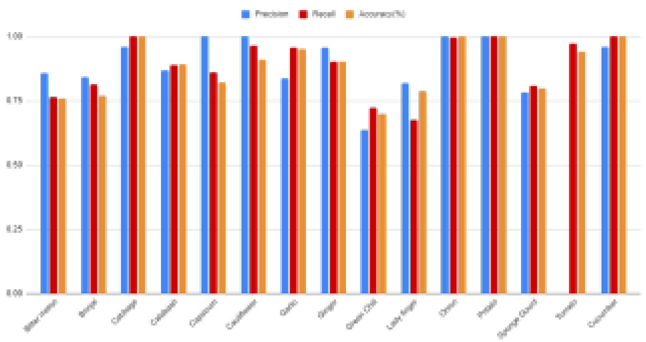

Fig. 5 Class vs Precision, Recall and Accuracy

Fig. 6 A. Predicted vegetables model

Fig. 6 B. Predicted vegetables model

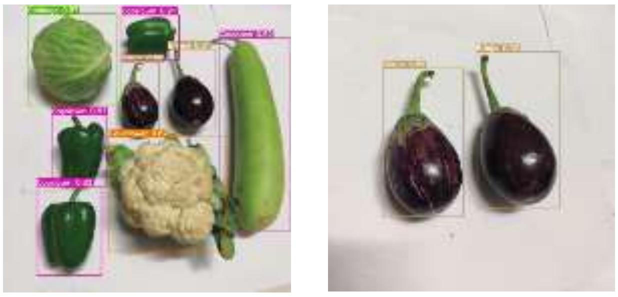

Fig. 6 C. Predicted vegetables model

Fig. 6 D. Predicted vegetables model

As shown in Figs. 5.a and 5.b, we manually tested the train model and obtained predictions with higher confidence scores. For each trial, we evaluated the performance of the model and trained it again until we got a good amount of accuracy, for which we increased and augmented the dataset.

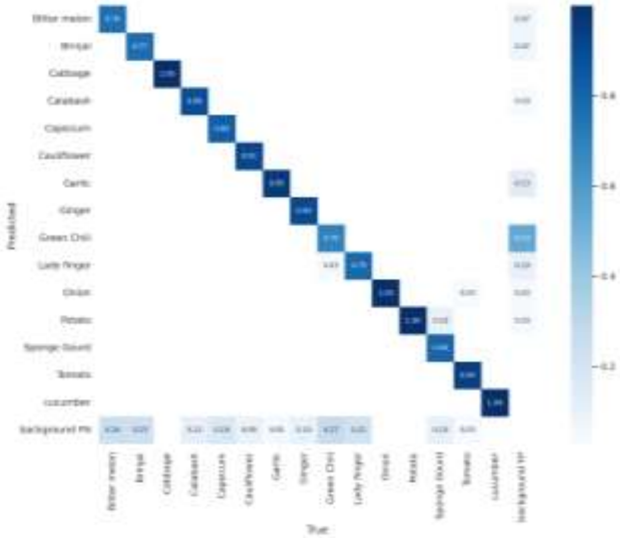

Fig. 7 Confusion Matrix

Confusion Matrix allows one to visualize the performance of the model over each unique class. It is a comparative plot of actuality and prediction. Most of the factors like precision, recall and accuracy are dependent on this tabular representation. As shown in Fig. 7 the model achieved higher value at true positive and and true negative in most of the classes.

Fig. 7 User Interface to add or take photo

Fig. 7 shows the user interface to take image input from the user. The Open Camera feature accesses the device's built-in camera to take photos, or users can upload images directly through the device.

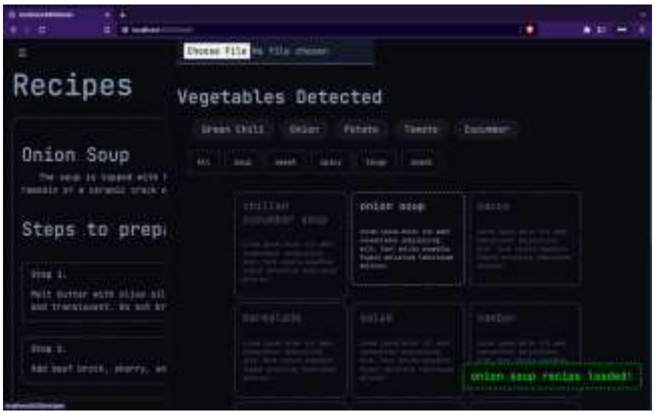

Fig. 8 User Interface after detection.

We built a web interface to provide section-wise functionality for the system. Fig. 8 shows the user interface after the detection. On the right, the user is presented with a list of detected vegetables, along with recipes, which are sorted based on the user's history. While the left-side section consists of the procedure to make the recipe.

To increase the response time, we remove the rewriting of the bounding box on the image after detection, which gives a faster response during the integration phase. Ajax caused the faster reloading of dynamic data, which reduced the extra loading of content.

V. CONCLUSION

This paper presents the Vision Based Intelligent Recipe Recommendation System (VBIRRS). People who stay remotely away from their homes although know how to cook, they may not know which recipes can be made out of available vegetables. This system is not only intelligent enough to recommend the recipes possible from the available vegetables also remembers the preferences of the user and recommends the recipes smartly. The system uses object detection algorithm YOLO and recommends the Indian recipes. We have achieved the classification accuracy more than 81.6%. We built a web interface to provide section wise functionality of the system. To reduce the response time, we remove the rewriting of the bounding box on image after detection which gives a faster response during integration phase. Ajax caused the faster reloading of dynamic data which reduced the extra loading of content.

REFERENCES

[1]. Yanai, Keiji, Takuma Maruyama, and Yoshiyuki Kawano. "A Cooking Recipe Recommendation System with Visual Recognition of Food Ingredients." International Journal of Interactive Mobile Technologies 8.2 (2014).

[2]. Hu, Xinyu, Yuheng Liu, and Yu Xing. "Food Ingredient Detection For Recipe Recommendation Systems."

[3]. Morol, Md Kishor, et al. "Food recipe recommendation based on ingredient detection using deep learning." Proceedings of the 2nd International Conference on Computing Advancements. 2022

[4]. Mokdara, Tossawat, Priyakorn Pusawiro, and Jaturon Harnsomburana. "Personalized food recommendation using deep neural network." 2018 Seventh ICT International Student Project Conference (ICT-ISPC). IEEE, 2018.

[5]. Sachin, C., et al. "Vegetable Classification Using You Only Look Once Algorithm." 2019 International Conference on Cutting-edge Technologies in Engineering (ICon-CuTE). IEEE, 2019.

[6]. Li, Zhengxian, et al. "A scalable recipe recommendation system for mobile application." 2016 3rd International Conference on Information Science and Control Engineering (ICISCE). IEEE, 2016.

[7]. Minaee, Shervin, et al. "Image segmentation using deep learning: A survey." IEEE transactions on pattern analysis and machine intelligence (2021).

[8]. Zhou, Lei, et al. "Application of deep learning in food: a review." Comprehensive reviews in food science and food safety 18.6 (2019): 1793-1811.

[9]. Maheshwari, Suyash, and Manas Chourey. "Recipe Recommendation System using Machine Learning Models." International Research Journal of Engineering and Technology (IRJET) 6.9 (2019): 366-369.

[10]. Gite, B. B., Aarti Nagarkar, and Chaitali Rangam. "Recommendation of Indian Cuisine Recipes Based on Ingredients."

[11]. Teng, Chun-Yuen, Yu-Ru Lin, and Lada A. Adamic. "Recipe recommendation using ingredient networks." Proceedings of the 4th annual ACM web science conference. 2012.

[12]. Lipi Shah, Hetal Gaudani, Prem Balani (2016), “Recipe Recommendation System using Hybrid Approach”, IJARCCE.

[13]. S.T Patil, Rajesh D. Potdar (2019), “Recipe Recommendation Systems”, IOSR Journals.

[14]. Wang, Chien-Yao, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for realtime object detectors." arXiv preprint arXiv:2207.02696 (2022).

[15]. Chou, Jian, et al. "A method of optimizing Django based on greedy strategy." 2013 10th web information system and application conference. IEEE, 2013.

[16]. Latha, R. S., et al. "Fruits and Vegetables Recognition Using YOLO." 2022 International Conference on Computer Communication and Informatics (ICCCI). IEEE, 2022.

[17]. Diwan, Tausif, G. Anirudh, and Jitendra V. Tembhurne. "Object detection using YOLO: Challenges, architectural successors, datasets and applications." Multimedia Tools and Applications (2022): 1-33.

[18]. Khedkar, Sonam, et al. "Real time databases for applications." International Research Journal of Engineering and Technology (IRJET) 4.06 (2017): 2078-2082

[19]. Mulani, A. O., & Mane, P. B. (2017). Watermarking and cryptography based image authentication on reconfigurable platform. Bulletin of Electrical Engineering and Informatics, 6(2), 181-187.

[20]. Mulani, A. O., Jadhav, M. M., & Seth, M. (2022). Painless Non- invasive blood glucose concentration level estimation using PCA and machine learning. the CRC Book entitled Artificial Intelligence, Internet of Things (IoT) and Smart Materials for Energy Applications.

[21]. Mane, P. B., & Mulani, A. O. (2019). High throughput and area efficient FPGA implementation of AES algorithm. International Journal of Engineering and Advanced Technology, 8(4).

[22]. Mandwale, A. J., & Mulani, A. O. (2015, January). Different Approaches For Implementation of Viterbi decoder on reconfigurable platform. In 2015 International Conference on Pervasive Computing (ICPC) (pp. 1-4). IEEE.

[23]. Kulkarni, P., & Mulani, A. O. (2015). Robust invisible digital image watermarking using discrete wavelet transform. International Journal of Engineering Research & Technology (IJERT), 4(01), 139-141.

[24]. Kashid, M. M., Karande, K. J., & Mulani, A. O. (2022, November). IoT-Based Environmental Parameter Monitoring Using Machine Learning Approach. In Proceedings of the International Conference on Cognitive and Intelligent Computing: ICCIC 2021, Volume 1 (pp. 43-51). Singapore: Springer Nature Singapore.

[25]. Godse, A. P., & Mulani, A. O. (2009). Embedded systems. Technical Publications.

[26]. Deshpande, H. S., Karande, K. J., & Mulani, A. O. (2015, April). Area optimized implementation of AES algorithm on FPGA. In 2015 International Conference on Communications and Signal Processing (ICCSP) (pp. 0010-0014). IEEE.

[27]. Mulani, A. O., & Mane, P. B. (2019). High-Speed area-efficient implementation of AES algorithm on reconfigurable platform. Computer and Network Security, 119.

[28]. Mulani, A. O., & Mane, P. B. (2014, October). Area optimization of cryptographic algorithm on less dense reconfigurable platform. In 2014 International Conference on Smart Structures and Systems (ICSSS) (pp. 86-89). IEEE.