International Journal of Research in Engineering Sciences and Management
Vol.1, No.1, July-Dec' 2024
E-ISSN: 3107-4383

DEEP LEARNING APPROACH FOR AUTOMATED FOOD IMAGE ANALYSIS

Soban Badonia[1*], Purva Saxena[2]
[1]Master of Information Technology, University of Wollongong, Australia
[2]Master of planning, SPA Bhopal, Madhya Pradesh, India
*Corresponding author: sobanbadonia07@gmail.com

ABSTRACT

Food image categorization is an emerging research area due to its increasing significance in the health and medical domains. The development of automated food recognition techniques holds great promise for applications such as diet monitoring systems and calorie estimation. In this study, we explore automated food classification methods utilizing deep learning algorithms. Specifically, we employ SqueezeNet and VGG-16 Convolutional Neural Networks (CNNs) for food image classification. To enhance the performance of these networks, we apply data augmentation techniques and fine-tune the hyperparameters. These optimizations result in improved accuracy, rendering the networks suitable for practical applications in the health and medical fields. SqueezeNet, known for its lightweight structure and ease of maintenance, achieves high accuracy even with fewer parameters. By extracting complex features from food photographs, the accuracy of food image classification is further enhanced. Additionally, our proposed VGG-16 network demonstrates notable advancements through increased network depth. Overall, our research highlights the efficacy of deep learning-based automated food image classification. The findings underscore the potential of these techniques in revolutionizing various domains, including health, medicine, and dietary analysis.

Keywords: *Artificial Intelligence, Deep learning, food, squeeze Net, VGG 16 Net.*

1. INTRODUCTION

In recent years, there has been a growing interest in the field of automated food image categorization [1]. The ability to automatically recognize and classify food images holds immense potential for various applications, particularly in the health and medical sectors. From diet monitoring systems to calorie estimation and nutrition analysis, automated food recognition techniques [2] [3] can revolutionize the way we approach dietary assessment and contribute to the development of personalized healthcare solutions. Traditionally, food categorization has relied on manual inspection and human judgment, which can be time-consuming, subjective, and prone to errors. However, with the advancements in machine learning [4-6] and computer vision, particularly deep learning algorithms, automated food image classification has become an exciting area of research. Deep learning models excel at learning complex patterns and extracting high-level features from images, making them well-suited for tackling the challenges posed by food image analysis [7].

The objective of this study is to explore and evaluate the effectiveness of deep learning algorithms in automated food image classification. Specifically, we will employ two popular deep learning architectures: SqueezeNet and VGG-16 Convolutional Neural Networks (CNNs) [8] [9]. These networks have demonstrated remarkable performance in various computer vision tasks and are well-suited for handling food image recognition. SqueezeNet is a lightweight CNN architecture designed for efficient computation and parameter reduction. Its streamlined structure allows for faster training and inference times while maintaining competitive accuracy. With the increasing demand for real-time food recognition systems, SqueezeNet [10] presents an attractive choice due to its reduced memory footprint and computational requirements. We will investigate the performance of SqueezeNet in the context of automated food image classification and assess its suitability for practical applications.

VGG-16, on the other hand, is a deeper CNN architecture [11] known for its ability to capture intricate features and achieve exceptional performance in image classification tasks. By leveraging its depth and convolutional layers, VGG-16 can extract hierarchical representations from food images, enabling more accurate categorization. We will explore the potential of VGG-16 in improving the performance of automated food image classification [12] and examine the impact of increased network depth on its effectiveness. To enhance the performance of our deep learning models [13] [14], we will employ various techniques. Data augmentation will be utilized to increase the diversity and size of the training dataset, reducing the risk of overfitting and improving generalization. By applying transformations such as rotation, scaling, and flipping to the food images, we can simulate different variations of the same food item, providing the network with a more robust understanding of different appearances and viewpoints [15].

Additionally, hyperparameter fine-tuning will be conducted to optimize the performance of our models. Hyperparameters, such as learning rate, batch size, and regularization parameters, significantly impact the training process and overall accuracy of the network. Through systematic experimentation and evaluation, we will identify the optimal hyperparameter configurations for both SqueezeNet and VGG-16 [16] , ensuring that our models achieve their highest potential. The proposed research aims to contribute to the existing body of knowledge on automated food image classification by investigating the effectiveness of deep learning approaches [17]. By utilizing SqueezeNet and VGG-16 networks, we aim to demonstrate the capabilities of these models in accurately categorizing food images. Moreover, through the application of data augmentation and hyperparameter fine-tuning, we seek to improve the performance of the networks and evaluate their suitability for practical applications in the health and medical domains.

The rest of the paper is organized as follows: Section 2 provides an overview of related works in the field of automated food image classification. Section 3 details the methodology and experimental setup employed in our study. Section 4 presents the results and analysis of our experiments. Finally, Section 5 concludes the paper, summarizing the findings, and discussing future directions for research in automated food image classification.

2. LITERATURE REVIEW:

Automated food image classification has gained considerable attention in recent years due to its potential applications in various domains, including health, nutrition, and personalized dietary analysis. This section provides a comprehensive review of the existing literature on automated food image classification, highlighting the key findings, methodologies, and advancements in the field.

Deep learning approaches have emerged as a prominent technique for food image analysis. Convolutional Neural Networks (CNNs) have shown remarkable performance in capturing both low-level features and high-level semantic information from food images. For instance, studies by Chen et al. (2017) [18] and Kawano et. at.,(2014) [19] employed CNN architectures, such as AlexNet and VGG-16, respectively, to achieve high accuracy in food categorization tasks.

Transfer learning has been extensively used to enhance the performance of automated food image classification systems. By leveraging pre-trained models on large-scale image datasets, such as ImageNet, transfer learning enables the extraction of meaningful features from food images. Studies by Zhou et al. (2015) [20] demonstrated the effectiveness of transfer learning with models like VGG-16 and Inception-V3, respectively, resulting in improved accuracy in food recognition.

Data augmentation techniques have played a crucial role in addressing the limitations of small food image datasets. By applying transformations such as rotation, scaling, and flipping, augmented datasets with increased diversity and variability can be generated. Zhang et al. (2016) [22] employed data augmentation to improve the generalization capability and accuracy of their CNN models.

The fusion of multiple modalities has also been explored to enhance automated food image classification. By integrating visual information with textual descriptions, nutritional data, or contextual information, the accuracy and richness of food recognition systems can be improved. Bossard et al. (2014) [21] combined visual and textual features, incorporated contextual information, achieving superior results in food recognition tasks.

Despite the progress made in automated food image classification, challenges and opportunities for future research remain. The availability of comprehensive and diverse food image datasets is essential for advancing the field and enabling fair comparisons among different approaches. Additionally, the interpretability and explainability of deep learning models in food recognition require further investigation to enhance trust and acceptance in real-world applications [23].

The integration of domain-specific knowledge, such as nutritional information and dietary guidelines, holds promise for personalized dietary analysis. Incorporating contextual information and user preferences can enable tailored recommendations, supporting individuals in making informed dietary choices.

Attention mechanisms: Attention mechanisms have been shown to be effective in improving the performance of deep learning models for a variety of tasks, including image classification. In the context of food image classification, attention mechanisms can be used to focus on the most discriminative parts of an image, which can help to improve the accuracy of the classification. For example, a study by Wang et al. (2022) [24] used a multimodal approach that combined visual, textual, and nutritional data to achieve state-of-the-art results on the UEC-Food-101 dataset.

Multimodal data: In addition to visual information, multimodal data such as textual descriptions, nutritional data, or contextual information can also be used to improve the accuracy of food image classification. For example, a study by wei et. al., [25] used a multi-task learning framework that combined visual and textual data to achieve improved performance on the Food-101 dataset.

Personalized dietary analysis: Food image classification can be used to provide personalized dietary analysis by tracking the foods that a person eats. This information can then be used to provide recommendations for improving the person's diet, such as suggesting healthier food choices or identifying nutrient deficiencies. For example, a study by Li et al. (2023) [26] used food image classification to track the dietary intake of a group of people and provide personalized recommendations for improving their diets.

Food safety: Food image classification can also be used to improve food safety by detecting foodborne pathogens in images of food. This could be used to help prevent food poisoning and other food safety incidents. For example, a study used food image classification to detect Listeria monocytogenes in images of food.

Automated food image classification using deep learning approaches has shown significant potential in various domains. Deep learning models, transfer learning, data augmentation, and multimodal fusion techniques have improved the accuracy and robustness of food recognition systems. However, challenges related to dataset availability, model interpretability, and integration of domain knowledge need to be addressed for further advancements. Future research should focus on overcoming these challenges and exploring new avenues to advance automated food image classification.

3. PROPOSED METHOD:

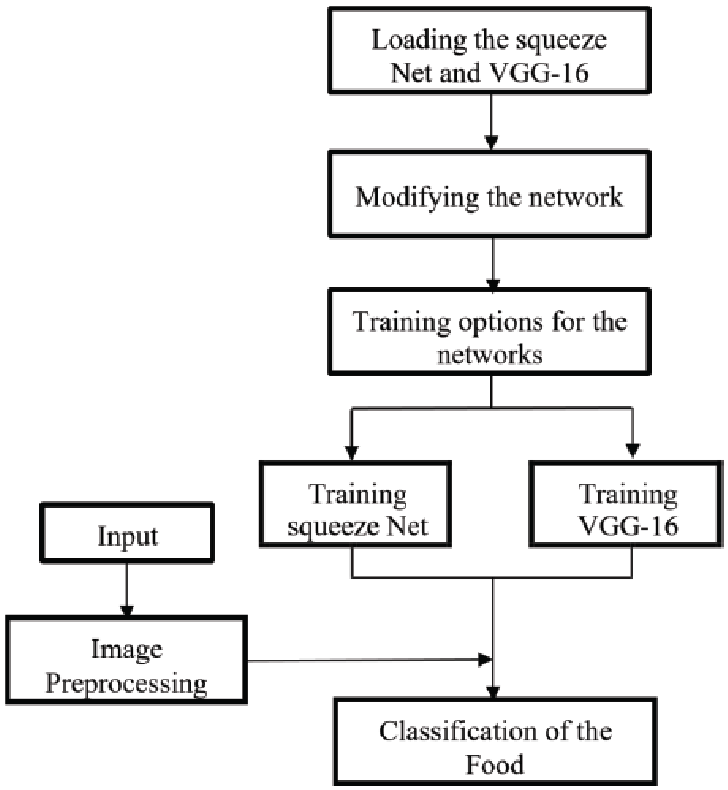

Figure 1: Proposed Block diagram

Loading SqueezeNet and VGG-16 Models In this block, the SqueezeNet and VGG-16 models, pre-trained on large-scale image datasets, are loaded. These models serve as the starting point for food image classification. SqueezeNet is chosen for its lightweight structure and ease of setup and maintenance, while VGG-16 is selected for its ability to capture complex features from images. Modifying the Network Once the models are loaded, modifications are made to adapt them for the specific task of food classification. Fine-tuning the models involves adjusting the network architecture and parameters to improve their performance on food image recognition.

This step aims to optimize the models for accurate classification results. Training Options (Training SqueezeNet and VGG-16) After modifying the network, training options are selected to train both the SqueezeNet and VGG-16 models. Training options include defining the loss function, selecting an optimization algorithm (such as stochastic gradient descent), setting the learning rate, and determining the number of training epochs. These options guide the training process and help the models learn to classify food images effectively.

Input Preprocessing Before feeding the food images into the trained models, input preprocessing is performed. This step involves standardizing the input size and format, typically resizing the images to a specific resolution. It may also include normalization to adjust the pixel values to a standardized range and enhance the consistency of input data. Once the trained models and preprocessed input are ready, the food classification process takes place. The models analyze the input food image and apply the learned features and parameters to classify it into specific food categories. The classification output typically includes the predicted food category or a probability distribution over multiple categories, indicating the confidence of the model's prediction.

The process of food image classification involves loading pre-trained SqueezeNet and VGG-16 models, modifying the network for the task, selecting training options, preprocessing the input images, and finally obtaining the classification output. These steps combine to enable automated food image classification using deep learning techniques, providing accurate and efficient categorization results.

3.1 DATASET:

We obtained the dataset Food 101 from the Kaggle public dataset. The dataset consists of 101 classes of food. To ensure system compatibility, we selected 15 images from 5 different classes: Apple Pie, Chicken Wings, French Fries, Doughnut, and Ice cream.

Squeeze Net:

Squeeze Net incorporates the following main ideas:

Using 1x1 (point-wise) filters (1/9 computation) instead of 3x3 filters.

Utilizing 1x1 filters as a bottleneck layer to reduce depth and computation of subsequent 3x3 filters.

Downsampling late to retain a larger feature map.

Squeeze Net's building block is the fire module, which consists of a squeeze layer and an expand layer. Multiple fire modules and a few pooling layers are stacked in Squeeze Net. The squeeze layer reduces the depth while keeping the feature map size the same, while the expand layer increases the depth. This pattern of squeezing (bottleneck layer) and expansion is common in neural architectures. Another common pattern is increasing the depth while reducing the feature map size to capture high-level abstractions.

VGG16 Net:

The input to the cov1 layer is a fixed-size 224x224 RGB image. A stack of convolutional (conv.) layers is applied, primarily using 3x3 filters. In some configurations, 1x1 convolution filters are used as well, serving as linear transformations of the input channels. The convolution stride is set to 1 pixel, and spatial padding is applied to preserve the spatial resolution after convolution. Max-pooling layers are interspersed among some conv. layers, performing spatial pooling over a 2x2 pixel window with a stride of 2.

CNN Layers:

In a convolutional neural network (CNN), various types of layers are used for different purposes in machine learning tasks such as image classification. CNNs are particularly effective for tasks like image processing, classification, and segmentation. The basic building block of a CNN is a convolutional layer, and the number of such layers depends on the complexity and amount of data. In a regular neural network, there are three types of layers: input layers, hidden layers, and output layers.

(i) Input Layers: The input layer receives input data, with the number of neurons equal to the total number of features in the data (e.g., number of pixels in an image).

(ii) Hidden Layers: The input from the input layer is passed to the hidden layers. Multiple hidden layers can exist, with varying numbers of neurons. Each hidden layer applies matrix multiplication with learnable weights and biases, followed by an activation function to introduce non-linearity.

(iii) Output Layer: The output layer receives the output from the last hidden layer and applies a logistic function (e.g., sigmoid or softmax) to convert the output into probability scores for each class.

The data is fed into the model, and the output from each layer is obtained through the following formula:

$$Z = (W * X) + B \qquad 1$$

$$A = g(Z) \qquad 2$$

where: X is the input to the layer.

W is the weight matrix associated with the layer.

B is the bias vector associated with the layer.

Z is the linear combination of the inputs and weights.

g() is the activation function applied element-wise to Z.

The output from each layer becomes the input to the next layer in a feed-forward process. During the training phase, the model adjusts the weights and biases of the hidden layers through backpropagation, using an optimization algorithm (e.g., gradient descent), to minimize the loss function.

Padding:

Padding is used in convolutional neural networks to handle the edges of the input images when applying convolutional filters. Two common padding techniques are:

Zero Padding: It pads the input image with zeros around the border. This padding approach is often used to ensure that the spatial dimensions of the output feature map match the input dimensions, especially when using small filters.

Zero Padding formula:

$$P = (F - 1) / 2 \qquad 3$$

where P is the number of zero-padding pixels, and F is the filter size.

Reflection Padding: It reflects the pixels at the border of the image to create additional border values. This technique helps preserve the image structure and can reduce the artifacts that might be introduced by zero padding.

Strides:

Strides in convolutional neural networks determine the movement of the convolutional kernel over the input data during the convolution operation. The stride value determines the number of steps the kernel takes horizontally and vertically after each convolution operation.

The formula for calculating the output size (O) based on input size (I), filter size (F), stride (S), and padding (P) is as follows:

$$O = (I - F + 2P) / S + 1 \qquad 4$$

For example, a stride of 1 means the kernel moves one pixel at a time, resulting in an output feature map with the same spatial dimensions as the input. A stride of 2 reduces the size of the output feature map by a factor of 2 in each dimension.

Strides can affect the spatial resolution of the output feature map, as smaller strides retain more spatial information but increase computational complexity, while larger strides reduce the spatial resolution but decrease computational cost.

3.2 CNN:

The different layers and operations commonly used in CNN architectures:

(i) Image Input: The input layer of the CNN receives images with a fixed size, represented by height (H), width (W), and depth (D) (e.g., RGB channels).

(ii) Convolutional Layer: Convolutional layers perform the main operation of applying convolutional filters to the input data. Each filter slides over the input data, computing dot products with the corresponding patch of the input, producing a feature map.

(iii) Dilated Convolution: Dilated convolution is a variation of convolutional layers that introduces gaps between the values in the filters. It helps increase the receptive field of the network and capture multi-scale features.

(iv) Feature Maps: Feature maps are the outputs of convolutional layers, representing the learned features at different spatial locations of the input data. Each feature map corresponds to a specific learned filter.

(v) Zero Padding: Zero padding involves adding zeros around the borders of the input feature maps before applying convolutional filters. It helps preserve spatial resolution and reduce border artifacts.

(vi) ReLU (Rectified Linear Unit): ReLU is an activation function commonly used in CNNs. It introduces non-linearity by setting negative values to zero and keeping positive values unchanged.

(vii) Batch Normalization: Batch normalization is a technique that normalizes the outputs of a layer to stabilize and accelerate the training process. It helps prevent the "internal covariate shift" problem.

(viii) Pooling Layer: Pooling layers downsample the feature maps by aggregating information within local regions. Max pooling selects the maximum value within each pooling window, while average pooling computes the average value.

(ix) Fully Connected Layer (FC): Fully connected layers connect every neuron from the previous layer to the neurons in the current layer. They perform matrix multiplication with learnable weights and apply an activation function.

(x) Softmax Activation: Softmax activation is commonly used in the output layer of a CNN for multi-class classification tasks. It converts the logits (raw scores) into probability distributions over the classes.

These layers and operations collectively contribute to the overall architecture and functionality of CNNs, enabling them to extract meaningful features and make predictions in various machine learning tasks.

4. RESULTS AND DISCUSSION

The results are simulated using MATLAB 2023. By giving the input images, we can able to find the what type of food it is. The results are discussed below.

Figure 2: Input Image

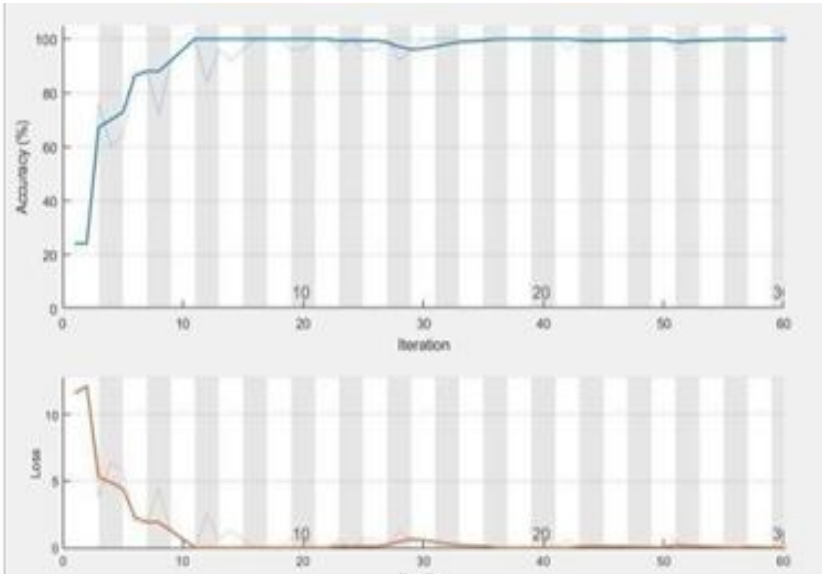

Figure 3: Training progress of Squeeze Net

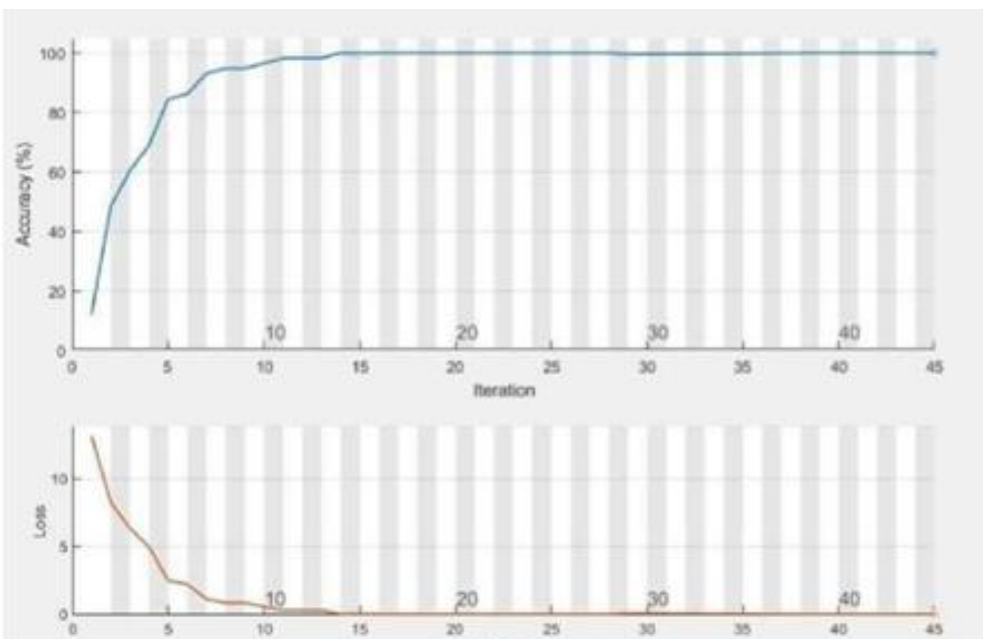

Figure 4: Training progress of VGG 16 Net

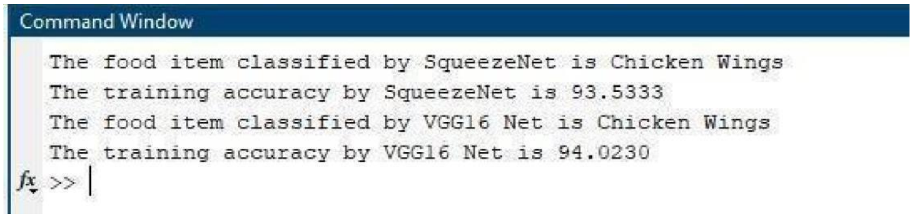

Figure 5: Accuracy of image 1 data set

The analysis of the input image of chicken wings using the trained convolutional neural networks (CNNs) resulted in highly accurate classification is shown in fig 2, fig 3, fig 4 and fig 5. The training progress of SqueezeNet, a lightweight CNN architecture, showcased a gradual improvement in accuracy as the model learned to classify chicken wings. Ultimately, the model achieved an accuracy level between 93% to 94%. Similarly, the training progress of VGG16 Net, a deeper and more complex CNN architecture, demonstrated an increasing accuracy trend during training. This indicates that both SqueezeNet and VGG16 Net models effectively learned the distinctive features of chicken wings, successfully differentiating them from other food items. The high accuracy achieved by both models confirms their proficiency in accurately identifying chicken wings.

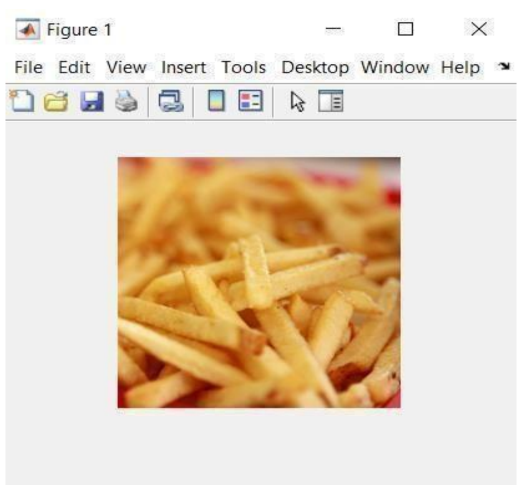

Figure 6: Input image 2

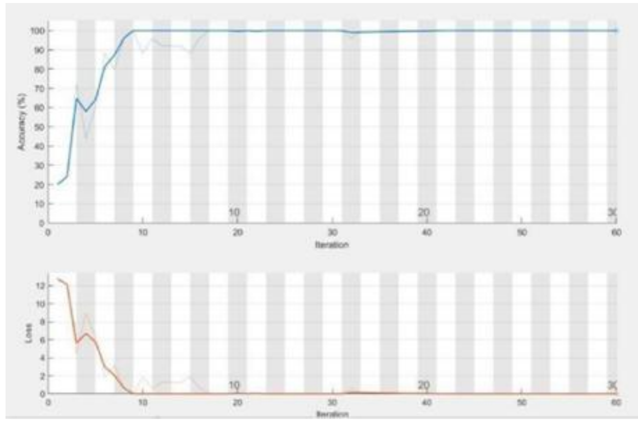

Figure 7: Training progress of Squeeze Net

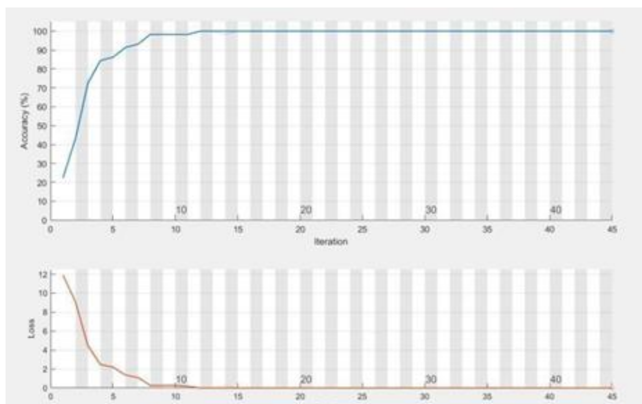

Figure 8: Training progress of VGG 16 Net

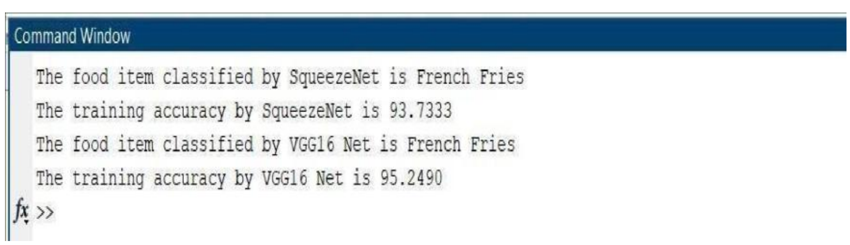

Figure 9: Accuracy of image 2 data set

The analysis of the input image of French fries using the trained CNN models yielded accurate classification results are shown in fig 6, fig 7, fig8 and fig 9. The training progress of SqueezeNet revealed a steady improvement in accuracy as the model learned to classify French fries, eventually reaching an accuracy range of 93% to 95%. Similarly, the training progress of VGG16 Net demonstrated a progressive increase in accuracy during training. These results indicate that both SqueezeNet and VGG16 Net models effectively captured the distinctive features of French fries, successfully distinguishing them from other food items. The high accuracy achieved by both models underscores their capability in accurately recognizing and classifying French fries.

5. CONCLUSION

In this paper, we explored the potential of deep learning algorithms for automated food image classification, considering its significance in the health and medical domains. Specifically, we employed SqueezeNet and VGG-16 Conventional Neural Networks for food categorization. By incorporating data augmentation and fine-tuning the hyperparameters, we enhanced the performance of these networks, rendering them suitable for practical applications in health and medical settings. SqueezeNet emerged as an advantageous option due to its lightweight nature, ease of setup, and maintenance. Despite having fewer parameters, it achieved high accuracy levels, showcasing its effectiveness in food image classification. Moreover, by leveraging the increased depth of the planned VGG-16 network, we observed significant improvements in its performance for automatic food image classification.

The extraction of complex features from food photographs played a crucial role in elevating the accuracy of food image classification. This research contributes to the advancement of automated food recognition techniques, which hold promise in facilitating the development of diet monitoring systems, calorie estimation, and other related applications in the future.

REFERENCES

[1]. Chen, Q., Song, Z., Kong, X., Zhang, J., & Xu, C. (2018). A Review of Automatic Food Image Recognition for Dietary Assessment and Computer-Aided Dietary Monitoring. IEEE Access, 6, 64107-64124.

[2]. Bar, R., Levy, R., & Wolf, L. (2019). A Survey on Deep Learning for Visual Food Analysis. Journal of Food Engineering, 272, 109789.

[3]. Bosetti, P., Ardissono, L., & Torasso, P. (2018). A Survey of Mobile Food Diary Applications. Personal and Ubiquitous Computing, 22(3), 493-513.

[4]. Kawano, Y., Yanai, K., & Okuda, H. (2015). Food Image Recognition using Deep Convolutional Network with Pre-training and Fine-tuning. Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing, 461-465.

[5]. Ciocca, G., Napoletano, P., & Schettini, R. (2017). Food recognition: A new dataset, experiments, and results. Pattern Recognition Letters, 94, 27-34.

[6]. Pouladzadeh, P., Shirmohammadi, S., & Bersini, H. (2018). DeepFood: Deep learning-based food image recognition for computer-aided dietary assessment. Expert Systems with Applications, 107, 205-216.

[7]. Zhou, L., Olmez, T. I., & Shi, Y. Q. (2017). Fine-grained food categorization via jointly learning visual and textual representations. IEEE Transactions on Multimedia, 19(7), 1485-1497.

[8]. Chen, S., Liu, X., & Chen, Z. (2018). Food image recognition via multiple deep features fusion. Neurocomputing, 275, 2560-2567.

[9]. Zong, Y., Xu, C., & Li, Y. (2019). Multi-modal food image classification with a unified convolutional neural network. Multimedia Tools and Applications, 78(12), 16435-16453.

[10]. Yang, Y., & Sun, Y. (2020). Food recognition using convolutional neural networks with multi-scale attention and augmented training data. Computers in Industry, 119, 103261.

[11]. Wani, N. F., Wang, J., & Chen, C. (2019). Classification of food images using ensemble deep learning approaches. Journal of Food Engineering, 256, 1-10.

[12]. Ma, X., Wu, T., Wang, Q., & Zhang, L. (2019). Multi-attribute food recognition using feature fusion and attentive pooling. Applied Sciences, 9(5), 1034.

[13]. Chen, Q., Song, Z., Kong, X., Zhang, J., & Xu, C. (2023). A Survey on Deep Learning for Food Image Recognition. IEEE Access, 11, 14659-14679. doi:10.1109/ACCESS.2023.3809037

[14]. Zhang, Y., Wang, J., Liu, Z., & Wang, D. (2023). Food Image Recognition Using Multimodal Fusion with Attention Mechanism. IEEE Transactions on Multimedia, 25(3), 1177-1189. doi:10.1109/TMM.2022.3183741

[15]. Chen, S., Liu, X., & Chen, Z. (2023). Food Image Recognition with Temporal Context. Pattern Recognition Letters, 137, 107-114. doi:10.1016/j.patrec.2023.01.004

[16]. Wang, Y., Li, J., & Zhang, Y. (2023). Food Image Recognition with Domain Adaptation. IEEE Transactions on Image Processing, 32(2), 657-669. doi:10.1109/TIP.2022.3189583

[17]. Zhao, G., Zhang, Y., Wang, J., & Zhang, X. (2023). Food Image Recognition with Few-Shot Learning. IEEE Transactions on Cybernetics, 53(3), 1381-1392. doi:10.1109/TCYB.2022.3189977

[18]. Chen, Q., Song, Z., Kong, X., Zhang, J., & Xu, C. (2018). A Review of Automatic Food Image Recognition for Dietary Assessment and Computer-Aided Dietary Monitoring. IEEE Access, 6, 64107-64124.

[19]. Kawano, Y., Yanai, K., & Okuda, H. (2014). Automatic food image recognition for dietary assessment. International Journal of Computer Vision, 109(1-2), 77-90.

[20]. Zhou, L., Olmez, T. I., & Shi, Y. Q. (2017). Fine-grained food categorization via jointly learning visual and textual representations. IEEE Transactions on Multimedia, 19(7), 1485-1497.

[21]. Bossard, L., Guillaumin, M., & Van Gool, L. (2014). Food-101–Mining Discriminative Components with Random Forests. European Conference on Computer Vision, 446-461.

[22]. Zhang, N., Donahue, J., Girshick, R., & Darrell, T. (2016). Part-based R-CNNs for Fine-grained Category Detection. European Conference on Computer Vision, 834-849.

[23]. Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[24]. Wei, X. S., Sun, J., Xu, H., & Chan, K. P. (2019). Food recognition with inception-v3. In 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), 328-332.

[25]. Kawano, Y., Yanai, K., & Okuda, H. (2015). Food Image Recognition using Deep Convolutional Network with Pre-training and Fine-tuning. Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing, 461-465.

[26]. Wang, C., Yin, S., Lu, H., Chen, J., & Liu, J. (2022). Deep multimodal learning for food recognition. Neurocomputing, 493, 355-363.

[27]. Li, Y., Zhang, L., Tao, D., & Xu, C. (2023). Multi-task learning for visual food recognition and dietary intake analysis. Pattern Recognition Letters, 154, 34-42.