Journal of Theoretical and Applied Information Technology

15th December 2023. Vol.101. No 23

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

REVOLUTIONIZING COMPUTER VISION: ENHANCED FOOD IMAGE CLASSIFICATION WITH SWIN TRANSFORMER AND SVM CLASSIFIER

ELPINA[1], GEDE PUTRA KUSUMA[2]

Computer Science Department, BINUS Graduate Program - Master of Computer Science,

Bina Nusantara University, Jakarta, Indonesia, 11480

E-mail: [1]elpina001@binus.ac.id, inegara@binus.edu

ABSTRACT

This paper presents a novel approach for food image classification using a combination of the Swin Transformer model and a support vector machine (SVM) classifier. The proposed method surpasses the performance of the original Swin Transformer model trained on ImageNet, achieving an impressive accuracy of 91.05% on the testing dataset. Comparative evaluation shows that the SVM classifier enhances the classification capabilities of the Swin Transformer, outperforming the baseline approach. The results highlight the efficacy of the Swin Transformer as a feature extraction model for food image classification tasks. The integration of deep learning with traditional machine learning techniques, as demonstrated by the SVM classifier, shows promise for improving classification accuracy in various applications such as food recognition systems and dietary analysis tools. Future work includes further optimization of the proposed method, exploring domain adaptation and transfer learning techniques, and investigating advanced fusion methods to achieve even higher classification accuracy and improved generalization across diverse food domains.

Keywords: *Food Image Classification, Deep Learning, Vision Transformer, Swin Transformer, Feature Extraction, Support Vector Machine*

1. INTRODUCTION

Food image classification has gained attention due to its broad applications, driven by social media, food blogging, and dietary monitoring systems. The demand for automated food item identification spans nutrition analysis, recipe recommendation, menu recognition, and quality inspection. Automatic classification of food images has the potential to revolutionize our interaction with food data. Advancements in deep learning models and large-scale datasets have enabled researchers to tackle the complexities of food recognition. State-of-the-art computer vision techniques are being leveraged to improve the accuracy and efficiency of food image classification. Accurately classifying food items from images is challenging due to variations in shape, color, texture, and size [1]. Lighting conditions during image capture can introduce inconsistencies, and intra-class variability adds further complexity [2]. Overcoming these challenges is crucial for unlocking the full potential of computer vision in food image analysis and enabling accurate and automated food recognition.

There is a research gap in utilizing the Swin Transformer [3], a cutting-edge vision Transformer [4], for feature extraction in food image analysis. While convolutional neural networks (CNNs) are extensively studied for this task, the Swin Transformer offers unique advantages, such as its hierarchical architecture and shifted windowing scheme, making it adaptable for image classification. However, its application for food image classification, particularly combined with Support Vector Machine (SVM) [5], is largely unexplored. This research aims to investigate the effectiveness of the Swin Transformer for feature extraction and its compatibility with SVM classifiers, advancing the field and providing insights into its potential for computer vision tasks. The main objectives are to assess the Swin Transformer's effectiveness in feature extraction for food image classification and evaluate the performance of an SVM classifier using these features. By applying the Swin Transformer to food images, this research aim to capture meaningful visual features to enhance food classification.

Additionally, this study seek to assess how the SVM classifier leverages these features for accurate classification. The primary goal is to demonstrate the potential of the Swin Transformer and SVM for improving the accuracy and effectiveness of food image classification.

In this study, the demonstration used the Food-101 [6] dataset, consisting of training and testing sets. The training set was used to optimize the Swin Transformer model's parameters, while the testing set evaluated the performance of both the model and the SVM classifier. This research's main contribution is utilizing the Swin Transformer for feature extraction in food image classification, combined with an SVM classifier. The study address the research gap in food image classification and advance computer vision techniques in this area. Our work demonstrates the Swin Transformer's effectiveness in capturing discriminative features from food images, improving classification accuracy. The integration of the SVM classifier provides a robust framework for food image classification. Our research has practical implications in dietary monitoring, nutrition analysis, and restaurant menu recognition. Accurate classification enables personalized nutrition recommendations, automated food tracking, and efficient menu recognition systems. Overall, this study bridge the gap between vision Transformer advancements and their real-world application, advancing the state-of-the-art in food image classification.

The primary contribution of this research lies in the utilization of the Swin Transformer model for feature extraction in food image classification, coupled with the application of an SVM classifier. By employing the Swin Transformer, which is a cutting-edge vision Transformer model, this research address the research gap in the domain of food image classification and contribute to the advancement of computer vision techniques in this specific area. Our work extends the current knowledge by demonstrating the effectiveness of the Swin Transformer in capturing discriminative features from food images, leading to improved accuracy in classification. Moreover, the integration of the SVM classifier provides a robust and interpretable framework for food image classification tasks. The outcomes of our research have practical implications in various domains, including dietary monitoring, nutrition analysis, and restaurant menu recognition. The accurate classification of food images can enable personalized nutrition recommendations, facilitate automated food tracking for dietary purposes, and enhance the efficiency of restaurant menu recognition systems. Overall, our research contributes to the field of food image classification by bridging the gap between the latest advancements in vision transformer models and their practical application in real-world scenarios, ultimately advancing the state-of-the-art in this domain.

The remaining sections of this paper are organized as follows: In Section 2, this exploration provide an overview of the related work on food image classification and deep learning models. Section 3 presents the theoretical background and concepts pertaining to the deep learning models employed in our study. In Section 4, this research cover the detailed methodology, including the implementation steps, architectural details of the Swin Transformer model, and SVM classifier. Section 5 is dedicated to presenting the results and analysis of our experiments, including a performance comparison against baseline methods. Finally, in Section 6, this study conclude the paper by summarizing the contributions of our research, discussing its limitations, and suggesting avenues for future work.

2. RELATED WORKS

In 2017, Pan et al. [7] proposes a framework called DeepFood for the classification of food ingredients using deep learning techniques. The researchers evaluate the DeepFood framework on a balanced multi-class dataset consisting of 41 classes of food ingredients, with 100 images for each class. They compare the proposed feature extractor with popular pre-trained CNN models, including AlexNet [8], CaffeNet [9], and ResNet [10]. Experimental results show that the deep features extracted using ResNet outperform the other models and achieve the highest average accuracy.

The authors also explore different feature evaluators, such as PCA [11], CFS [12], and IG, and use ranking metrics to select the best deep feature subsets. They compare various benchmark classifiers (Random Forest [13], Bagging, BayesNet) with the SMO classifier [14] used in the DeepFood framework. The experiments demonstrate that the DeepFood framework, integrating ResNet deep feature sets, IG feature selection, and the SMO classifier, outperforms other techniques for food-ingredients recognition. The best model achieves an average accuracy of 87.78%.

The paper concludes by emphasizing the effectiveness of the proposed DeepFood framework for multi-class classification of food ingredients. It combines the advantages of ResNet deep feature sets, IG feature selection, and the SMO classifier to

significantly improve classification accuracy. The research results show a substantial improvement over existing methods and provide a high-performance solution for the automatic classification of food ingredients using deep learning.

In 2018, to address the growing concern of obesity and its related health conditions, McAllister et al. [15] explored automated food image classification for dietary monitoring. They applied pretrained ResNet-152 [16] and GoogLeNet [17] convolutional neural networks (CNNs) to extract deep features from various food image datasets, including Food 5K [18], Food-11, RawFooT-DB [19], and Food-101 [6]. The extracted features were then used to train machine learning classifiers such as Artificial Neural Networks (ANN), SVM, random forests, fully connected neural networks, and naive Bayes.

The results showed that utilizing ResNet-152 deep features with SVM and a food-101,g (RBF) kernel achieved high accuracy, with 99.4% accuracy for Food-5K dataset. Additionally, ANN [20] trained with ResNet-152 features achieved 91.34% and 99.28% accuracy for Food-11 and RawFooT-DB datasets, respectively. For the more challenging Food-101 dataset, ResNet-152 features yielded a moderate accuracy of 64.98% with SVM and RBF kernel [21]. The study demonstrated the effectiveness of deep CNN features in diverse food item classification tasks, with ResNet-152 consistently achieving higher accuracies across multiple datasets.

This research highlighted the potential of using deep learning features extracted from pretrained CNNs for food image classification. It compared the performance of ResNet-152 and GoogLeNet and found that ResNet-152 features consistently outperformed in various datasets. Moreover, the authors discussed the generalization power of ResNet-152 features and the efficiency of using generic deep features for binary classification tasks like food and non-food classification. The study provided insights into the performance of different machine learning classifiers and emphasized the convenience of combining deep learning with traditional machine learning approaches for effective image classification.

In 2020, to develop a feature extraction system and fusion scheme based on the characteristics of Asian food, Wu, Zhao, & Qu [22] proposed a food image classification model using the SLGC (SURF-Local and Global Color) [23] technique which combines image segmentation and feature fusion. This system builds a feature representation method that combines SURF features, local color information and global color information, which can extract Asian food image features comprehensively and efficiently. At the same time, an image segmentation algorithm is used to separate invalid interference information and highlight the food subject, further enhancing the image classification effect. To reduce the influence of food image backgrounds on feature extraction, SLGC uses the GrabCut [24] algorithm to segment food images.

Caltech 101 [25] and UEC FOOD 100 [26] were used as datasets in their experiments. Caltech101 is an integrated image dataset with rich content and a wide variety of image types. UEC FOOD 100 is a popular Japanese food, which can fully reflect the structural characteristics of Asian food. The experimental results show that SLGC based on fusion features can effectively improve the image classification effect with a classification accuracy of about 64%.

Razali et al. [27] compared the performance of 70 combinations of food recognition approaches, consisting of six different CNN-based pre-trained models used as feature extractors, one feature representation based on the RGB component of the image, and ten machine learning classifiers used often used for. In addition, two types of datasets are used for performance evaluation, namely the Sabah Food Dataset [27] and the VIREO-Food172 [28] dataset. In the Sabah Food Dataset, it was found that the EFFNet [29] + CNN approach gave the best performance with an accuracy of 94.01%, followed by Xception [30] + SVM (OVO) with an accuracy of 86.32%. The significant decrease in accuracy from 94.01% to 86.32% indicates that there may be outliers in the EFFNet + CNN model so that it only works well on certain training and testing datasets from the Sabah Food Dataset and does not represent the best overall approach. Whereas in the VIREO-Food172 Dataset, it was found that EFFNet + SVM (OVO) gave the best performance with an accuracy of 86.57%, followed by EFFNet + LSVM [31] (OVO) with an accuracy of 85.60%. Compared to the Sabah Food Dataset, the difference between the best performing and second-best performing approaches in the VIREO-Food172 Dataset is insignificant at 0.97%.

It can be concluded that the best feature representation for the Sabah Food and VIREO-Food172 dataset is the feature representation based on EFFNet. This is supported by the discussion of the paper on feature representation Overall Score, which shows that EFFNet has the highest feature representation Overall Score. A similar comparison was made for classifiers, and it was found that the LSVM (OVO) classifier provided the best

performance for food recognition, followed by LSVM (OVO) as the classifier. In terms of computational complexity and memory space usage, Xception with 2048 feature dimensions, can be considered for a small reduction in accuracy performance.

3. THEORY

In the context of this research, computer vision plays a crucial role in automating the analysis and interpretation of visual data for efficient and accurate classification tasks. By harnessing the power of computer vision techniques, such as feature extraction and pattern recognition, this study aims to address the challenges of image classification by exploring the potential of the Swin Transformer model combined with a SVM classifier. By leveraging computer vision capabilities, this research seeks to improve the accuracy and efficiency of image classification systems, contributing to advancements in fields like medical imaging, autonomous driving, and video surveillance.

3.1 Swin Transformer

The Swin Transformer introduces a hierarchical design and a shifted window approach, which allows it to model at various scales, accommodating the variations in the scale of visual entities. It constructs hierarchical feature maps by starting with small-sized patches and gradually merging neighboring patches in deeper Transformer layers. This hierarchical representation enables the Swin Transformer to leverage advanced techniques for dense prediction tasks like object detection and semantic segmentation [3].

The shifted window approach in the Swin Transformer brings efficiency by limiting self-attention computation to non-overlapping local windows. It computes self-attention within these windows, reducing the computational complexity to linear with respect to the image size. The shifted windows also provide connections among them, enhancing the modeling power of the Transformer. This approach is more efficient in terms of real-world latency compared to sliding window methods used in previous self-attention-based architectures. The Swin Transformer's hierarchical design, shifted window approach, and linear computational complexity make it a suitable general-purpose backbone for computer vision applications. Swin Transformer offers a competitive speed-accuracy trade-off. It provides an alternative architecture that can complement or replace CNNs as the backbone network in vision tasks and encourages the exploration of unified modeling between vision and language signals [32].

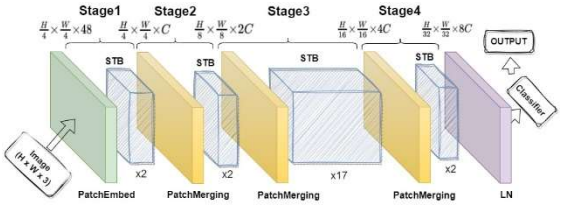

Figure 1: Swin Transformer Architecture

Figure 1 illustrates the initial stage of the Swin Transformer architecture. The input RGB image is divided into non-overlapping patches using a patch splitting module, similar to ViT [4]. Each patch is treated as a token and its feature is obtained by concatenating the raw pixel RGB values. The patch size is set to 4 × 4, resulting in a feature dimension of 48 (4 × 4 × 3). A linear embedding layer is applied to project this raw-valued feature into an arbitrary dimension denoted as C. The patch tokens are then processed by several Swin Transformer blocks, which involve modified self-attention computations. These Transformer blocks maintain the number of tokens ($H/4 \times W/4$) and, along with the linear embedding, are referred to as "Stage 1".

To create a hierarchical representation, the number of tokens is gradually reduced through patch merging layers as the network goes deeper. The first patch merging layer concatenates the features of each group of neighboring 2 × 2 patches and applies a linear layer to the concatenated features, resulting in a dimension of 2C. This reduces the number of tokens by a factor of 4 (2× down sampling of resolution) and defines the output dimension as 2C. Subsequently, Swin Transformer blocks are applied to transform these features, while maintaining a resolution of (H/8) × (W/8). This initial combination of patch merging and feature transformation is referred to as "Stage 2". The same procedure is repeated twice for "Stage 3" and "Stage 4", resulting in output resolutions of H/16 × W/16 and H/32 × W/32, respectively. As a result, the proposed architecture can conveniently replace the backbone networks in existing methods for various vision tasks [33].

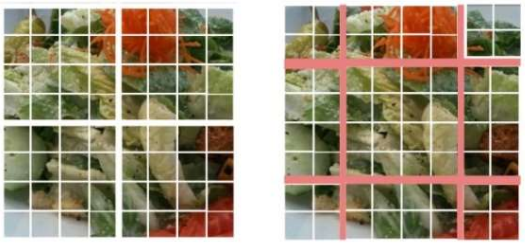

Figure 2: Shifted window method on Swin Transformer

In Swin Transformer blocks, self-attention is applied within specific windows, while merging occurs in the final feature map. This approach effectively addresses the computational challenges associated with increasing image sizes in traditional ViT models. However, since self-attention is limited to fixed windows, the relationships between windows are not fully captured. To overcome this limitation, the windows are shifted to the right (→) and down (↓) by half the window size, enabling the calculation of the relationship between two windows through an additional round of self-attention. As a result, the Swin Transformer allows for comprehensive analysis of the entire input image using self-attention within individual windows. In summary, the Swin Transformer, as depicted in Figure 2, ensures a thorough examination of the input image through self-attention within individual windows, overcoming the limitations of fixed windows in capturing inter-window relationships [3].

The Swin Transformer is designed for detection and segmentation tasks by incorporating hierarchical feature maps and shifted windows into the ViT model. Unlike conventional transformers that use tokens with the same patch size for self-attention, the Swin Transformer gradually merges adjacent patches, starting from a patch size of 4x4. This approach resembles the hierarchical structure of the feature pyramid network, enabling the utilization of information from each hierarchical feature map, similar to the U-Net architecture [34].

3.2 SVM Classifier

SVM is a powerful supervised machine learning algorithm commonly used for classification tasks. SVM operates by creating an optimal hyperplane that separates different classes in the feature space. The objective is to find the hyperplane that maximizes the margin between classes, allowing for better generalization and robustness to new data points. The SVM algorithm works by transforming the input data into a higher-dimensional feature space using a kernel function. This transformation enables the SVM to effectively handle nonlinear classification problems by finding a linear decision boundary in the transformed space. Commonly used kernel functions are linear, polynomial, and RBF. SVM learns from a training dataset by identifying support vectors, which are the data points that determine the location of the separating hyperplane. These support vectors play a crucial role in defining the decision boundary and contribute to the overall classification accuracy [5].

In this case, the classifier is trained using the popular RBF kernel and linear kernel. Linear kernel is commonly used when the data is linearly separable, meaning it can be separated by a single line, whereby its function takes a linear form. Where RBF kernel excels at capturing complex non-linear relationships in the data. Each kernel Linear and RBF function can be seen as follows respectively:

$$K(x_i, x_j) = x_i, T\, x_j \qquad (1)$$

$$K(x, y) = exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right) \qquad (2)$$

In this study, the parameter C is used to balance the trade-off between minimizing training error and maintaining a broader margin in SVM models. By adjusting C, the focus can be shifted towards fitting the training data accurately or promoting better generalization to unseen data. A smaller C value results in a larger margin and higher tolerance for misclassified training instances, aiming for a more robust and generalized model. On the other hand, a larger C value prioritizes accurate classification of all training instances, leading to a tighter decision boundary. To determine the optimal C value, grid search techniques are employed, evaluating the model's performance on various validation sets. This approach facilitates unbiased assessment across different C values and enables the selection of the value that strikes the desired balance between training accuracy and generalization capability [5].

4. METHODS

In this section, this exploration will explain about the experiments carried out, from dataset, image pre-processing, training, validation, and model testing.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4.1 Dataset

Figure 3: Example of images in the Food-101 Dataset

The Food-101 [6] dataset is widely recognized as a prominent benchmark dataset for image classification tasks specifically focused on food recognition. It comprises a diverse collection of 101 distinct food categories, each containing 1,000 images. These images were sourced from various online recipe websites and contributed by both professional and amateur photographers, resulting in a rich and challenging dataset. The images exhibit varying resolutions and aspect ratios, with the shortest side ranging from 384 to 512 pixels. A visual representation of images from the Food-101 Dataset can be observed in Figure 3.

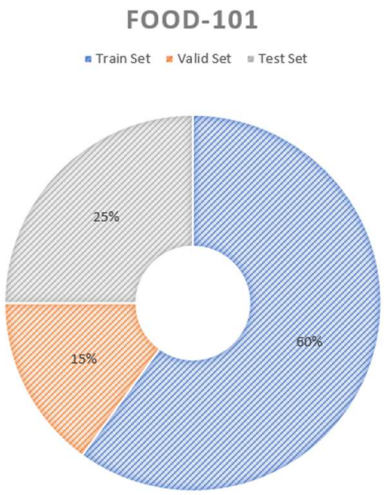

Figure 4: Division of the Food-101 dataset

Each image in the dataset is associated with a single label representing one of the 101 food categories. These categories encompass a wide range of popular dishes, including "apple pie," "carrot cake," "hamburger," "pizza," "sushi," and many more. The dataset provides a predefined train/test split, where 75% of the images are designated for training purposes, while the remaining 25% are reserved for evaluation. For a visual representation of this dataset configuration, please refer to Figure 4.

4.2 Pre-Processing

In the preprocessing step, the input images are transformed to ensure consistency and facilitate effective model training. Two common transformations applied to the images are resizing and data augmentation. The first transformation resizes the images to a uniform size of 224x224 pixels. This resizing operation helps standardize the image dimensions across the dataset, which is beneficial for training deep learning models that require inputs of consistent sizes. The second transformation refers to data augmentation techniques applied to the images. Data augmentation is a widely used approach to artificially increase the size of the training dataset by applying various random transformations to the images. These transformations can include random rotations, translations, flips, brightness adjustments, and more. By augmenting the training data with these variations, the model learns to be more robust and generalizable to different image conditions and variations present in real-world scenarios.

Figure 5: Example of augmented images

Figure 5 presents examples of randomly augmented images, showcasing differences in factors such as lighting, clarity, food item orientations, and more. The combination of resizing and data augmentation techniques helps improve the model's ability to learn meaningful features and patterns from the images while increasing its robustness to variations and potential overfitting. These transformations contribute to enhancing the performance and generalization capability of the model during the training process.

4.3 Model Architecture

This study propose utilizing the Swin Transformer model for feature extraction from the training dataset. Following this extraction, the obtained features are inputted into an SVM classifier for training. Subsequently, the trained model, which combines the Swin Transformer and SVM, is employed to predict the testing data. This testing data has undergone feature extraction using the Swin model as well. The architecture of our proposed method is visually illustrated in the accompanying Figure 6.

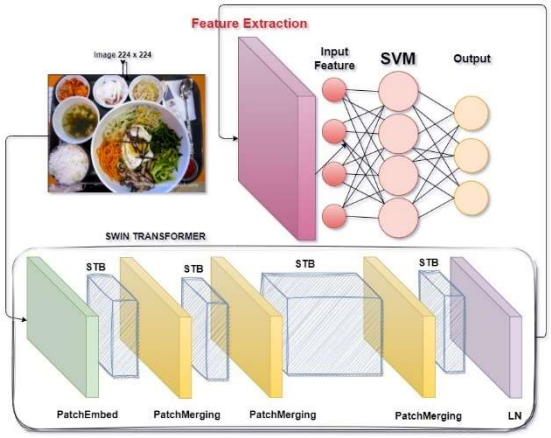

Figure 6: The proposed method architecture : Swin Transformer feature extraction combined with SVM classifier

4.3.1 Swin transformer training

The proposed method in this research involves the implementation of the Swin Transformer model using the swin_base_patch4_window7_224 variant on the Food-101 dataset. The input images are partitioned into patches, representing local regions, and linearly embedded to create patch tokens. These tokens are passed through multiple connected Swin Transformer blocks, which consist of attention and feed-forward layers. This allows the model to capture global context and model relationships between patches. In subsequent stages, patch merging and down-sampling operations are performed, reducing spatial dimensions and increasing the channel dimension. This hierarchical process is repeated multiple times, enabling the model to capture features at different scales.

During the training process of the Swin Transformer model, a batch size of 64 was chosen to strike a balance between improved gradient estimation and computational efficiency. This decision considers the available computational resources and ensures that the model can effectively process a reasonable number of training examples simultaneously. To measure the discrepancy between the model's predictions and the ground truth labels, the FlattenedLoss function, which is based on CrossEntropyLoss, is used. This loss function quantifies the model's performance and guides the parameter updates to minimize prediction errors.

To optimize the model's parameters during training, the Adam optimizer is employed. The Adam optimizer offers the advantage of adaptively adjusting the learning rates for each parameter based on their historical gradients. This adaptive optimization approach enables the Swin Transformer model to converge faster and potentially overcome local optima during training. By dynamically adjusting the learning rates, the Adam optimizer helps the model efficiently navigate the optimization landscape and improve its classification performance. The EarlyStoppingCallback technique will also be applied to avoid model underfitting or overfitting during training. With this combination, it is hoped to achieve good results in the training or fine-tuning of the model within a reasonable timeframe.

Backpropagation and gradient descent algorithms also optimize the model's parameters, minimizing the loss and updating the weights. The training process is performed iteratively over three epochs, with the entire training dataset processed in mini batches. Validation on a separate dataset is conducted regularly to monitor the model's performance and detect overfitting.

4.3.2 Feature Extraction

Feature extraction is performed by passing images from the Food-101 dataset through the Swin Transformer model and capturing the output from a specific layer as representative features. The Swin Transformer architecture constructs hierarchical feature maps by aggregating patches at deeper layers. These feature maps exhibit linear computational complexity when dealing with varying image sizes due to the self-attention

calculations occurring only within localized windows. Leveraging this hierarchical feature map, the Swin Transformer model seamlessly incorporates advanced techniques for dense predictions, such as Feature Pyramid Networks (FPN), thereby serving as a backbone for sophisticated image classification and recognition tasks. In the implementation, the experiment employed a patch size of 4 × 4, resulting in a feature dimension of 4 × 4 × 3 = 48 for each patch. Linear embedding layers are applied to the raw features to project them into arbitrary dimensions.

In the implementation, the experiment will apply the technique of "head" truncation to the model. The concept of "head" truncation in CNN models refers to the removal of the last layer responsible for classification or prediction. These layers are typically designed to map learned representations to specific output classes. By truncating the "head" of the model, this exploration retain the underlying layers responsible for extracting high-level features from input data. These underlying layers capture crucial spatial and semantic information from the input data, which proves highly valuable for various tasks such as feature visualization, transfer learning, or integration with other classifiers.

After removing the "head," the evaluation can pass input data through the modified CNN model and obtain the output from the preserved last layer. This output represents a set of features that encode relevant information about the input data. These extracted features can then serve as input for other machine learning algorithms, such as SVM, Random Forest, or even other neural networks. This strategy allows us to leverage the transformed CNN model for various downstream tasks while capitalizing on the meaningful features encoded within the retained layers.

4.3.3 SVM Classifier

The training process for the SVM classifier using Dask-ML [35] incorporates several steps to optimize performance and select suitable hyperparameters. The SVM classifier is a robust classification algorithm, and its effectiveness depends on hyperparameter choices and the kernel function employed. To handle large datasets efficiently, Dask-ML is employed, leveraging the parallel computing capabilities of Dask. The input data is chunked into manageable pieces using Dask arrays, enabling parallel processing. The GridSearchCV class from Dask-ML is used for hyperparameter tuning. It conducts an exhaustive search over the provided C values and evaluates each combination through cross-validation. Accuracy is selected as the evaluation metric for the SVM classifier's performance.

During training, the SVM classifier is fitted with the training data for each C value. Subsequently, predictions are generated on the test data, and the accuracy score is computed using Dask-ML's accuracy_score metric. The results demonstrate that the highest accuracy of 90.90% while trained on RBF kernel is achieved when C=1. While trained on Linear kernel, get the higher accuracy of 89.33%, more than 1% lower. This finding suggests that RBF kernel with a lower regularization strength leads to better generalization on the given dataset, indicating the SVM classifier's improved performance.

4.3.4 Performance Metrics

The performance of the proposed method is assessed using various evaluation metrics to gain insights into its effectiveness. , the performance of the Swin Transformer model is assessed using metrics such as training loss, validation loss, error rate, top-1 accuracy, and top-5 accuracy. Furthermore, Swin Transformer performance will be comprehensively presented through matrices recorded at each epoch, learning curves to visualize loss changes, and a classification report that provides additional insights into the model's performance. Accuracy can be computed by dividing the total correctly predicted data by the total predicted data. It provides a general overview of the model's predictive accuracy and is suitable for balanced datasets. The accuracy score is calculated by comparing the predicted labels with the ground truth labels and determining the percentage of correct predictions.

Top-1 accuracy measures the proportion of data where the output label is correctly predicted among all target labels. On the other hand, top-5 accuracy assesses the proportion of correct predictions where one of the top 5 initial predictions accurately matches the target label. Subsequently, training loss serves as a metric to evaluate how well the deep learning model fits the training data. It is computed by summing the error values for each data point in the training set. Similarly, validation loss, computed after each epoch, sums the error values for each data point in the validation set, though it does not contribute to weight updates. Validation training is calculated after each batch, while validation loss is measured after each epoch.

Furthermore, the classification of the machine learning model will be measured using accuracy, precision, recall, and F1-score. Accuracy is a metric that gauges how accurately the model classifies food

images. Precision assesses the model's ability to correctly identify food images within the intended category. Recall evaluates the model's ability to identify all relevant food images from the intended category. F1-score, a combined measure of precision and recall, provides a comprehensive overview of the model's performance.

```
GPU 0: NVIDIA A100-SXM4-40GB (UUID: GPU-a27e8601-30bb-ec61-f7e5-d1732752654b)
Model name:                      Intel(R) Xeon(R) CPU @ 2.20GHz
CPU(s):                          12
Thread(s) per core:              2
Core(s) per socket:              6
Socket(s):                       1
CPU MHz:                         2200.140
Available memory:                82G
Available disk:                  144G
```

Figure 7: Hardware Configuration

In this study, we evaluated the performance of swin transformer combined with SVM classifier using the GPU infrastructure of Google Colab for both training and evaluation. The GPU model utilized in the experiments was the NVIDIA A100-SXM4-40GB, a high-performance GPU engineered for accelerated computational tasks. The CPU employed in the experiments was the Intel(R) Xeon(R) CPU @ 2.20GHz, boasting a total of 12 cores with 2 threads per core, resulting in a total of 24 threads. The CPU operates at a clock speed of 2200.184 MHz. The system's available memory is reported at 82G, and the available storage space is reported at 144G. In terms of software, the experiments utilized various software libraries and frameworks. The code snippet provided includes the installation of the Dask-ML library, which is used for distributed machine learning tasks.

5. RESULT AND DISCUSSION

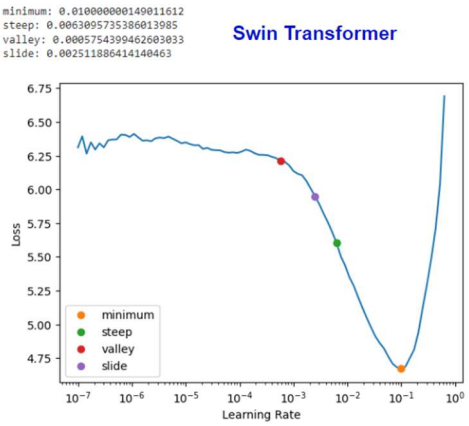

Figure 8: Learning rate for Swin Transformer

In the implementation of the Swin Transformer model, the initial learning rate value was set to 1e-3. Figure 8 illustrates the variations in loss values corresponding to changes in the learning rate – encompassing the minimum, steep, valley, and slide points. The valley point, characterized by a learning rate value around 0.00057, was selected as the optimal learning rate for the fine-tuning process. This choice aids the model in converging more effectively and enhancing performance in food recognition tasks.

epoch	train_loss	valid_loss	error_rate	accuracy	top_k_accuracy	time
0	1.185449	0.714394	0.193729	0.806271	0.953531	07:33

Better model found at epoch 0 with valid_loss value: 0.714393675327301.

epoch	train_loss	valid_loss	error_rate	accuracy	top_k_accuracy	time
0	0.951761	0.609924	0.164752	0.835248	0.963168	09:52
1	0.848331	0.539417	0.147657	0.852343	0.966337	09:52
2	0.681101	0.503993	0.137030	0.862970	0.969901	09:52
3	0.624669	0.473498	0.127591	0.872409	0.973399	09:52
4	0.620359	0.450272	0.120396	0.879604	0.973927	09:51
5	0.519798	0.443800	0.119010	0.880990	0.974587	09:52
6	0.498853	0.436171	0.118350	0.881650	0.974984	09:52
7	0.461342	0.429735	0.115182	0.884818	0.975116	09:52
8	0.456155	0.429031	0.114719	0.885281	0.975512	09:52
9	0.444791	0.427032	0.114257	0.885743	0.975446	09:52

Figure 9: Training results for Swin Transformer

Based on the results depicted in Figure 9, the training outcomes for the Swin Transformer model revealed a training loss of 0.444791 and a validation loss of 0.427032. The error rate was recorded at 0.114257, indicating the model's proficiency in minimizing misclassifications. Notably, the top-1 accuracy achieved an impressive 88.57%, showcasing the model's capability to accurately predict the primary class label.

Additionally, the top-5 accuracy which signifies the model's proficiency in identifying the correct class among the top five predictions, achieved an outstanding value of 97.54%. The training process was completed in approximately 9 minutes and 52 seconds. These results underscore the Swin Transformer's effectiveness in image classification tasks, demonstrating its robustness in capturing complex patterns and features within the data.

Furthermore, the evaluated model is subjected to testing using an unfamiliar testing dataset, consisting of 25,250 images, equivalent to 25% of the entire Food-101 dataset. The Swin Transformer model achieved a low loss value of 0.362943, indicating its effective learning and predictive capabilities. Furthermore, the error rate was impressively minimized to 0.101069, underscoring the model's proficiency in accurate classification. The model exhibits a testing accuracy performance

of 89.89%. These outcomes affirm the Swin Transformer model's robustness and efficacy in accurately classifying food images on previously unseen testing data.

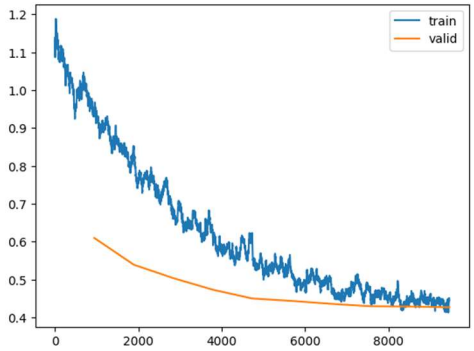

Figure 10: Plot loss graph

Figure 10 illustrates the graphs depicting the changing values of train loss and valid loss, affirming the balanced training and validation processes of the Swin Transformer model without signs of underfitting or overfitting.

Figure 11: Model evaluation examples

The satisfactory performance achieved by the Swin Transformer model is further highlighted in Figure 11. In the ninth example images, the model ensures that the predicted label matches the original label.

Table 1: SVM Classifier Result

Kernel	C	Accuracy
RBF	1	91,05
RBF	10	90,86
RBF	20	90,75
RBF	100	90,75
Linear	1	89,39
Linear	10	89,39
Linear	20	89,39
Linear	100	89,39

This study further leverages the Swin Transformer to extract features from the Food-101 dataset. The resultant features are subsequently trained by an SVM algorithm. The classification model training process will leverage Dask-ML, involving several steps to optimize performance and select appropriate hyperparameters. To efficiently handle large datasets, Dask-ML will be utilized, harnessing the parallel computing capabilities of Dask. Input data will be divided into manageable chunks using Dask arrays, enabling parallel processing. The GridSearchCV class from Dask-ML is employed for hyperparameter tuning. The training process using GridSearch CV will comprehensively search for given C values and evaluate each combination through cross-validation. Subsequently, predictions will be made on the test data, and accuracy scores will be computed based on the accuracy_score metric from Dask-ML.

As depicted in Table 1, the search encompassed both RBF and Linear kernels, with a range of C values. The highest accuracy was achieved when the SVM algorithm employed the RBF kernel with a C value of 1, resulting in a testing accuracy of 91.05%. The classification report from the feature classification using the SVM algorithm with the optimal parameters is presented in Figure 4.8. This report illustrates the precision, recall, and F1-score for each class, concluding with the average accuracy.

6. CONCLUSION

The proposed method, which combines the Swin Transformer model with an SVM classifier for feature extraction and classification, yielded promising experimental results. The Swin Transformer model was first fine-tuned on the training set of the Food-101 dataset using the fastai library, with ten epochs of training. This initial training process achieved a validation accuracy of 88.57%, indicating the model's capability to learn and generalize from the dataset. After fine-tuning the Swin Transformer model, the classifier head was removed to extract features from both the training

and test sets. These extracted features were then used as input for an SVM classifier. The SVM model was trained using the training set features and subsequently used to predict labels for the test set features.

The experimental results showed that the proposed method achieved a testing accuracy of 89.89% when using the Swin Transformer model alone for predictions. However, by leveraging the extracted features from the Swin Transformer model and utilizing an SVM classifier, the testing accuracy improved to 91.05%. This demonstrates the effectiveness of the combined approach in enhancing the classification performance. The obtained results highlight the complementary nature of the Swin Transformer model and the SVM classifier. The Swin Transformer's ability to capture rich and meaningful features from the images, combined with the discriminative power of the SVM classifier, led to improved classification accuracy. This showcases the potential of utilizing deep learning models in conjunction with traditional machine learning algorithms for improved performance in complex classification tasks.

The findings of our study, wherein we combined the Swin Transformer model with an SVM classifier for food image classification, align with and extend the insights provided by previous research in the field. The work of Pan et al. [7] and McAllister et al. [15] has established the effectiveness of utilizing deep learning features, particularly those extracted from models like ResNet-152, for food image classification tasks. Our approach builds upon this foundation, demonstrating the synergy between the Swin Transformer's feature extraction capabilities and the discriminative power of the SVM classifier. This resonates with the observations made by Wu, Zhao, & Qu [22], who emphasized the importance of comprehensive feature extraction techniques for food image classification. Moreover, our results corroborate with Razali et al. [27], showcasing the significance of combining advanced feature representation, such as that derived from the Swin Transformer, with robust classifiers like SVM for enhanced accuracy in food recognition tasks. The consistent improvement achieved by our combined approach underlines the potential for integrating deep learning models with traditional machine learning algorithms for more robust and accurate image classification, as discussed across various studies in the domain.

7. FUTURE WORK

In future research endeavors, a thorough exploration of hyperparameter optimization for the Swin Transformer model implementation is imperative. Leveraging frameworks like Optuna can facilitate the search for optimal parameter combinations, enhancing accuracy in data training and refining the classification of food images. Furthermore, there is a need to assess various Swin Transformer variants, gauging their performance in food image classification tasks using the Food-101 dataset. A comprehensive examination and comparison of these variants will yield insights into their distinct strengths and limitations.

Additionally, extending the investigation to encompass diverse machine learning classification algorithms, such as Artificial Neural Network, Naïve Bayes, K-Nearest Neighbors, is vital. Experimenting with these algorithms offers opportunities to evaluate and compare their efficacy in the nuanced context of food image classification. This comprehensive exploration aims to significantly contribute to the advancement of more effective and widely applicable food image classification methodologies. Further recommendations include evaluating the Swin Transformer model on alternative datasets to validate its generalization capabilities and considering its implementation on culturally specific food datasets for a more nuanced analysis.

REFERENCES

[1] P. Pouladzadeh and S. Shirmohammadi, 'Mobile Multi-Food Recognition Using Deep Learning', *ACM Trans. Multimed. Comput. Commun. Appl.*, vol. 13, no. 3s, pp. 1–21, Aug. 2017, doi: 10.1145/3063592.

[2] K. Hameed, D. Chai, and A. Rassau, 'Class distribution-aware adaptive margins and cluster embedding for classification of fruit and vegetables at supermarket self-checkouts', *Neurocomputing*, vol. 461, pp. 292–309, Oct. 2021, doi: 10.1016/j.neucom.2021.07.040.

[3] Z. Liu *et al.*, 'Swin Transformer: Hierarchical Vision Transformer using Shifted Windows', in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada: IEEE, Oct. 2021, pp. 9992–10002. doi: 10.1109/ICCV48922.2021.00986.

[4] A. Dosovitskiy *et al.*, 'An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale'. arXiv, Jun. 03, 2021. Accessed: May 21,

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

2023. [Online]. Available: http://arxiv.org/abs/2010.11929

[5] W. S. Noble, 'What is a support vector machine?', *Nat. Biotechnol.*, vol. 24, no. 12, pp. 1565–1567, Dec. 2006, doi: 10.1038/nbt1206-1565.

[6] L. Bossard, M. Guillaumin, and L. Van Gool, 'Food-101 – Mining Discriminative Components with Random Forests', in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2014, pp. 446–461. doi: 10.1007/978-3-319-10599-4_29.

[7] L. Pan, S. Pouyanfar, H. Chen, J. Qin, and S.-C. Chen, 'DeepFood: Automatic Multi-Class Classification of Food Ingredients Using Deep Learning', in *2017 IEEE 3rd International Conference on Collaboration and Internet Computing (CIC)*, San Jose, CA: IEEE, Oct. 2017, pp. 181–189. doi: 10.1109/CIC.2017.00033.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, 'ImageNet Classification with Deep Convolutional Neural Networks', in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2012. Accessed: Aug. 13, 2023. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html

[9] Y. Jia *et al.*, 'Caffe: Convolutional Architecture for Fast Feature Embedding'. arXiv, Jun. 20, 2014. doi: 10.48550/arXiv.1408.5093.

[10] K. He, X. Zhang, S. Ren, and J. Sun, 'Deep Residual Learning for Image Recognition', presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778. Accessed: Aug. 13, 2023. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html

[11] F. Kherif and A. Latypova, 'Chapter 12 - Principal component analysis', in *Machine Learning*, A. Mechelli and S. Vieira, Eds., Academic Press, 2020, pp. 209–225. doi: 10.1016/B978-0-12-815739-8.00012-2.

[12] M. A. Hall, 'Correlation-based feature selection for machine learning', Thesis, The University of Waikato, 1999. Accessed: Aug. 13, 2023. [Online]. Available: https://researchcommons.waikato.ac.nz/handle/10289/15043

[13] Y. Liu, Y. Wang, and J. Zhang, 'New Machine Learning Algorithm: Random Forest', in *Information Computing and Applications*, B. Liu, M. Ma, and J. Chang, Eds., in Lecture Notes in Computer Science, vol. 7473. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 246–252. doi: 10.1007/978-3-642-34062-8_32.

[14] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan, 'Multiple kernel learning, conic duality, and the SMO algorithm', in *Proceedings of the twenty-first international conference on Machine learning*, in ICML '04. New York, NY, USA: Association for Computing Machinery, Jul. 2004, p. 6. doi: 10.1145/1015330.1015424.

[15] P. McAllister, H. Zheng, R. Bond, and A. Moorhead, 'Combining deep residual neural network features with supervised machine learning algorithms to classify diverse food image datasets', *Comput. Biol. Med.*, vol. 95, pp. 217–233, Apr. 2018, doi: 10.1016/j.compbiomed.2018.02.008.

[16] Z. Wu, C. Shen, and A. van den Hengel, 'Wider or Deeper: Revisiting the ResNet Model for Visual Recognition', *Pattern Recognit.*, vol. 90, pp. 119–133, Jun. 2019, doi: 10.1016/j.patcog.2019.01.006.

[17] A. Singla, L. Yuan, and T. Ebrahimi, 'Food/Non-food Image Classification and Food Categorization using Pre-Trained GoogLeNet Model', in *Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management*, Amsterdam The Netherlands: ACM, Oct. 2016, pp. 3–11. doi: 10.1145/2986035.2986039.

[18] A. Şengür, Y. Akbulut, and Ü. Budak, 'Food Image Classification with Deep Features', in *2019 International Artificial Intelligence and Data Processing Symposium (IDAP)*, Sep. 2019, pp. 1–6. doi: 10.1109/IDAP.2019.8875946.

[19] C. Cusano, P. Napoletano, and R. Schettini, 'Evaluating color texture descriptors under large variations of controlled lighting conditions', *JOSA A*, vol. 33, no. 1, pp. 17–30, Jan. 2016, doi: 10.1364/JOSAA.33.000017.

[20] A. K. Jain, J. Mao, and K. M. Mohiuddin, 'Artificial neural networks: a tutorial', *Computer*, vol. 29, no. 3, pp. 31–44, Mar. 1996, doi: 10.1109/2.485891.

[21] X. Ding, J. Liu, F. Yang, and J. Cao, 'Random radial basis function kernel-based support vector machine', *J. Frankl. Inst.*, vol. 358, no. 18, pp. 10121–10140, Dec. 2021, doi: 10.1016/j.jfranklin.2021.10.005.

[22] R. Wu, S. Zhao, and Z. Qu, 'An SLGC Model for Asian Food Image Classification', *J. Comput. Commun.*, vol. 8, no. 4, Art. no. 4, Mar. 2020, doi: 10.4236/jcc.2020.84003.

[23] A. G. Delavar, 'SLGC: A New Cluster Routing Algorithm in Wireless Sensor Network for Decrease Energy Consumption', *Int. J. Comput. Sci. Eng. Appl.*, vol. 2, no. 3, pp. 39–51, Jun. 2012, doi: 10.5121/ijcsea.2012.2304.

[24] C. Rother, V. Kolmogorov, and A. Blake, '"GrabCut": interactive foreground extraction using iterated graph cuts', *ACM Trans. Graph.*, vol. 23, no. 3, pp. 309–314, Aug. 2004, doi: 10.1145/1015706.1015720.

[25] L. Fei-Fei, R. Fergus, and P. Perona, 'Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories', in *2004 Conference on Computer Vision and Pattern Recognition Workshop*, Jun. 2004, pp. 178–178. doi: 10.1109/CVPR.2004.383.

[26] Y. Matsuda, H. Hoashi, and K. Yanai, 'Recognition of Multiple-Food Images by Detecting Candidate Regions', in *2012 IEEE International Conference on Multimedia and Expo*, Jul. 2012, pp. 25–30. doi: 10.1109/ICME.2012.157.

[27] M. N. Razali *et al.*, 'Indigenous Food Recognition Model Based on Various Convolutional Neural Network Architectures for Gastronomic Tourism Business Analytics', *Information*, vol. 12, no. 8, Art. no. 8, Aug. 2021, doi: 10.3390/info12080322.

[28] J. Chen and C. Ngo, 'Deep-based Ingredient Recognition for Cooking Recipe Retrieval', in *Proceedings of the 24th ACM international conference on Multimedia*, in MM '16. New York, NY, USA: Association for Computing Machinery, Oct. 2016, pp. 32–41. doi: 10.1145/2964284.2964315.

[29] I. Freeman, L. Roese-Koerner, and A. Kummert, 'Effnet: An Efficient Structure for Convolutional Neural Networks', in *2018 25th IEEE International Conference on Image Processing (ICIP)*, Oct. 2018, pp. 6–10. doi: 10.1109/ICIP.2018.8451339.

[30] F. Chollet, 'Xception: Deep Learning With Depthwise Separable Convolutions', presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1251–1258. Accessed: Aug. 13, 2023. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2017/html/Chollet_Xception_Deep_Learning_CVPR_2017_paper.html

[31] O. L. Mangasarian and D. R. Musicant, 'Lagrangian Support Vector Machines', *J. Mach. Learn. Res.*, vol. 1, no. Mar, pp. 161–177, 2001.

[32] S. H. Lee, S. Lee, and B. C. Song, 'Vision Transformer for Small-Size Datasets'. arXiv, Dec. 26, 2021. Accessed: May 27, 2023. [Online]. Available: http://arxiv.org/abs/2112.13492

[33] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, and R. Timofte, 'SwinIR: Image Restoration Using Swin Transformer', in *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, Montreal, BC, Canada: IEEE, Oct. 2021, pp. 1833–1844. doi: 10.1109/ICCVW54120.2021.00210.

[34] N. Siddique, S. Paheding, C. P. Elkin, and V. Devabhaktuni, 'U-Net and Its Variants for Medical Image Segmentation: A Review of Theory and Applications', *IEEE Access*, vol. 9, pp. 82031–82057, 2021, doi: 10.1109/ACCESS.2021.3086020.

[35] M. Rocklin, 'Dask: Parallel Computation with Blocked algorithms and Task Scheduling', presented at the Python in Science Conference, Austin, Texas, 2015, pp. 126–132. doi: 10.25080/Majora-7b98e3ed-013.