Evaluating the Performance of CNNs for Food Classification with Data Augmentation and Transfer Learning

Sajad Shokoohifard

***Abstract*—Food classification is a challenging task in computer vision due to the large variability in the appearance and shape of different types of food items. Deep learning techniques, particularly convolutional neural networks (CNNs), have shown promising results in automating this task by automatically learning features from raw image data. In this study, we investigate and compare the performance of three different deep learning models for food classification: a simple CNN model (CNN1), a CNN model with data augmentation (CNN20), and a transfer learning-based model (TL45) using InceptionResNetV2 architecture. We evaluate the models on a food-11 dataset and compare their accuracy, precision, recall, and F1-score. Our experimental results show that the transfer learning-based model outperforms the other two models with a test accuracy of 85%.**

***Index Terms*—Food Classification, CNN, TL, InceptionResNetV2, food-11 dataset**

I. INTRODUCTION

FOOD classification is a fundamental task in the field of computer vision and has practical applications in various areas such as food industry, health, and nutrition. The classification of food items involves identifying different types of food items from images, and this task has been traditionally performed by human experts. However, with the recent advancements in deep learning techniques, it is now possible to automate the food classification process by training deep neural networks to recognize food items from images.

Deep learning (DL) techniques, particularly convolutional neural networks (CNNs), have been shown to achieve state-of-the-art performance in various computer vision tasks, including food classification. These techniques have the ability to automatically learn features from raw image data, which can then be used to accurately classify different types of food items. In recent years, many deep learning-based food classification models have been proposed in the literature, and these models have achieved impressive results on large-scale food classification datasets.

A. Motivation

It is difficult to classify food because there is a significant amount of variation in the way different types of food look and are shaped. Traditional machine-learning approaches require hand-engineered features, which are time-consuming and may not generalize well to new types of food items. DL techniques, on the other hand, can automatically learn features from raw image data and have shown promising results in food classification.

In this study, we investigate and compare the performance of three different deep learning models for food classification: a simple CNN model (CNN1), a CNN model with data augmentation (CNN20), and a transfer learning-based model (TL45). The aim is to provide insights into the effectiveness of these models for food classification and identify the most suitable approach for scenarios that require less time to train.

B. Main Contributions

The aim of this study is to compare the performance of three different DL models for food classification, providing insights into the strengths and weaknesses of each approach.

The CNN1 model serves as a baseline for comparison, while the CNN20 model aims to improve the performance of the baseline model by increasing the size of the training set and reducing overfitting. The TL45 model, on the other hand, leverages the pre-trained weights of a large-scale image classification model to achieve high accuracy with limited training data.

Overall, the study aims to contribute to the ongoing research in deep learning-based food classification by providing a comprehensive comparison of different approaches and identifying the most suitable approach for different scenarios.

C. Report Organization

The report is organized in the following way: In Section II, background information is provided on the topic of the study, including previous research on food classification. Section III provides an overview of the database and methodology used in the study. Section IV presents the experimental results. Finally, in Section V, the study is concluded.

II. LITERATURE REVIEW

A. Background

In short, Machine Learning (ML) is the practice of teaching machines to learn and improve through experience, which has driven significant advances in Artificial Intelligence (AI) in recent years. DL is a branch of ML that uses neural networks to enable machines to perform tasks, recognize patterns, and predict outcomes without human intervention, similar to the human brain. CNN is a supervised neural network that uses

Sajad Shokoohifard is with the Department of Electrical and Computer Engineering, University of Calgary (e-mail: sajad.shokoohifar1@ucalgary.ca)

perceptions to perform cognitive tasks, making it highly effective for image classification by extracting spatial and temporal dependencies. TL is a widely used ML technique that reuses pre-trained DL models to perform similar tasks, saving time by only training the classifier part of the model, and making it useful for small datasets. In this study, CNN is proposed for effective pixel data processing in image recognition and TL is recommended to reduce training time due to the relatively small dataset.

B. State of the art

[1] presents a study on the use of DL models for the automatic classification of food ingredients. The authors proposed a model, which is based on CNN architecture. The model is trained on a large dataset of food images and evaluated in terms of multi-class classification accuracy. The study also investigates the impact of different hyperparameters and architectures on the performance of the model.

[2] presents a study on using CNNs for food classification and ingredient estimation on a Raspberry Pi3 device. The authors proposed a food classification model based on a CNN architecture as an algorithm to estimate the ingredients present in the food based on the image. The study evaluates the performance of the proposed model and algorithm on the Raspberry Pi3 device in terms of accuracy, speed, and memory usage.

[3] investigates the use of deep TL techniques for food image classification. The authors proposed a framework that utilizes a pre-trained deep neural network model as a feature extractor and fine-tunes the model on a smaller dataset of food images. They compared the performance of their framework against other state-of-the-art methods and demonstrated that TL can significantly improve the accuracy of food image classification.

In [4], an approach to instance segmentation of visible cloud images was presented using the Mask R-CNN architecture and TL. The proposed method employs a pre-trained Mask R-CNN model trained on the COCO dataset and fine-tunes it on a small dataset of visible cloud images. The authors evaluated their approach on a dataset of visible cloud images and reported promising results in terms of instance segmentation accuracy.

In [5], the authors focused on developing a lightweight food image classification system specifically for Egyptian cuisine. A new dataset of Egyptian food images was proposed, and they trained and tested different models to classify the food images using TL techniques. They also evaluated the performance of the models on a low-power embedded system.

In [6], the authors proposed a method for food image classification using deep features. They utilized pre-trained deep neural networks to extract features from food images, which were then used to train and test different classifiers. The performance of their approach was evaluated on two datasets of food images and compared against other state-of-the-art methods.

[7] presents a novel approach to enhancing the accuracy of brain tumor segmentation in magnetic resonance imaging (MRI) scans using adaptive TL. The proposed method leverages pre-trained models from different domains and adapt them to the specific domain of brain tumor segmentation.

[8] investigates the effectiveness of fine-tuning pre-trained DL models for image classification tasks. The authors conducted experiments on the CIFAR-10 dataset using four different pre-trained models and evaluate the performance of the fine-tuning approach in terms of accuracy and convergence rate. The study provides insights into the optimal fine-tuning strategies for different pre-trained models and sheds light on the potential benefits and limitations of this approach.

[9] presents a study on the application of TL for weather image recognition using the Xception deep learning model. The author fine-tuned the pre-trained Xception model on a weather image dataset and evaluates its performance in terms of accuracy and computational efficiency. The study demonstrates that the fine-tuned Xception model outperforms other state-of-the-art models and achieves high accuracy in weather image recognition tasks.

In [10], a study on the use of DL and TL techniques was presented for skin cancer segmentation and classification. The authors proposed a multi-task learning framework that combines both segmentation and classification tasks and leverages a pre-trained ResNet50 model for TL. The proposed framework is evaluated on the ISIC 2018 dataset and compared with other state-of-the-art models in terms of accuracy and computational efficiency.

[11] presents a comparative study between two DL models, InceptionResnetV2 and InceptionV3, for attention-based image captioning. The authors fine-tuned both models on the MS-COCO dataset and evaluate their performance in terms of captioning accuracy and computational efficiency. The study also investigates the impact of different attention mechanisms on the performance of the models. The results show that InceptionResnetV2 outperforms InceptionV3 in terms of accuracy and that the attention mechanism significantly improves the performance of both models.

[12] presents a study on the use of the InceptionResNetV2 model for the classification of plant leaf diseases. The authors fine-tuned the InceptionResNetV2 model on a dataset of plant leaf images and evaluate its performance in terms of classification accuracy. The study also investigates the impact of different hyperparameters on the performance of the model. The results show that the InceptionResNetV2 model achieves high accuracy in plant leaf disease classification and outperforms other state-of-the-art models.

III. METODOLOGY

The process begins with processing the food-11 dataset that was taken from Kaggle; in that model input split into train, validation, and test sets in a ratio of 0.8:0.1:0.1. The models are then trained and validated using the same training and validation datasets. Finally, the trained models are tested on unseen testing data to achieve high prediction accuracy and low loss.

A. Pre-processing

The dataset, food-11, we used contains 16,000 food images divided among 11 classes, namely, Bread, Dairy product, Dessert, Egg, Fried food, Meat, Noodles-Pasta, Rice, Seafood, Soup, Vegetable-Fruit.

In addition, CNN1 model undergoes Pixel Normalization using Rescale. RGB images used in this study have pixel values that are integers between 0 and 255. Using these values as input for training could result in instability and an inability to learn during training since the model would treat each image differently. Normalization addresses this issue, which scales each pixel value to a range between 0 and 1 after normalization.

To minimize the randomness of the networks, a fixed seed number and batch size are utilized for all models. Careful consideration is given to selecting the batch size as the optimizer's ability to minimize the loss function is highly dependent on it. If the batch size is too large, the optimizer will be unable to minimize the loss function, while a small batch size will require a long time to reach the optimal solution as the number of iterations in each epoch depends on the batch size. The chosen batch size for this study strikes a fair balance between the quality of the result and the time taken to achieve it.

B. Food Classification Models

1) Model 1-CNN1: The model consists three layers. The first layer is a Conv2D layer with 16 filters, a kernel size of 3x3, and a ReLU activation function. The input shape is defined as (250,250,3) which corresponds to the image height, width, and the number of channels (RGB). The output of this layer is then passed through a MaxPooling2D layer with a pool size of 2x2. The second and third layers are similar to the first one but with more filters, 32 and 64 respectively. After each Conv2D layer, a MaxPooling2D layer is applied to downsample the feature maps. Then, a Flatten layer is added to convert the 2D feature maps into a 1D vector. This output vector is then fed into two fully connected (Dense) layers with 100 and 11 neurons, respectively. The activation function for the first dense layer is ReLU, while the second one uses the softmax activation function, which returns a probability distribution over the classes. Two dropout layers with a rate of 0.2 are added to refrain from overfitting the model.

The model is then compiled with a categorical cross-entropy loss function, Adam optimizer, and accuracy metric for evaluation during training.

The number of epochs for training is set to 12. Two callbacks are defined to be used during training. The callbacks are used to pass the list of defined callbacks to the fit method for monitoring and controlling the training process. The first callback saves the best model during training based on validation loss. The second callback is used as an early stopping callback, which stops training if the validation loss does not improve after a patience number of epochs, where patience is set to 5 in this case. Finally, the model's performance is evaluated on the validation and test sets. Fig.1 shows some data samples processed for the CNN1 model.

2) Model 2-CNN20: CNN20 is almost similar to CNN1, except for the addition of data augmentation, which aims to enhance its performance. Data augmentation is used to prevent overfitting. The augmentation techniques used were 'shear', 'zoom', 'rotation', 'height and width shift', 'horizontal flip', and 'fill mode'. These techniques helped to create a diverse perspective view of the image and keep the image dimensions the same. Some of the values used for augmentation were: shear and zoom range = 0.2, rotation range = 20, fill mode = nearest, and height and width shift range = 0.2.

Fig. 1. Sample of the processed dataset for CNN1 and CNN20

3) Model3-TL45: In this approach, a DL model using InceptionResNetV2 as the base architecture for image classification is exploited. The model is initialized with pre-trained weights from the ImageNet dataset and has the top layers removed. A custom top layer is added to the model to make classification on the target classes. The output of the base model is then flattened and passed through a fully connected layer with 100 neurons and the ReLU activation function. The output layer consists of a Dense layer with a softmax activation function with 11 units for allocating the class probabilities. The layers of the base model are frozen to prevent them from being trained during the training process.

The model is trained for 15 epochs. It is also using two callbacks - ModelCheckpoint and EarlyStopping - to save the best model during training and to stop training early if the validation loss does not improve after a specified number of epochs. The best model is then saved. Finally, it evaluates the model's performance on the validation and test sets. Fig.2 shows some data samples processed for the TL45 model.

C. Performance Analysis

To evaluate the models' performance, the confusion matrix is demonstrated for the three models(Fig.3, Fig.4, Fig.5). The precision, recall, F1-score, and support are computed from the matrices for each class. As can be seen the diagonal values for the TL45 model are higher which means the model is performing better in correctly classifying the instances.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

The study was conducted using Google Colaboratory, a Jupyter notebook environment designed for ML research and

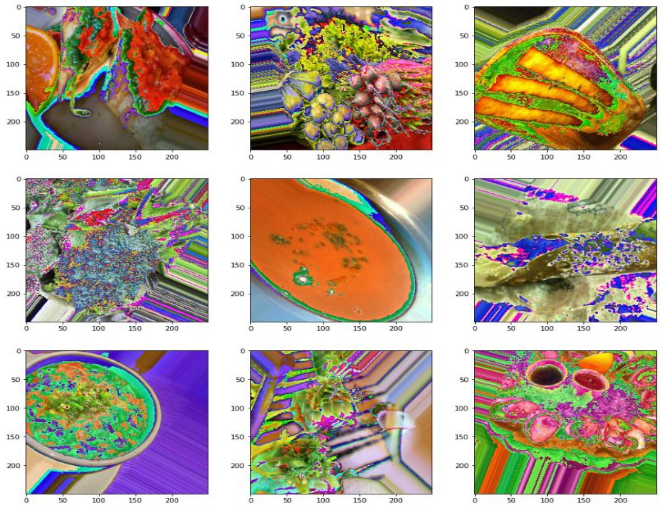
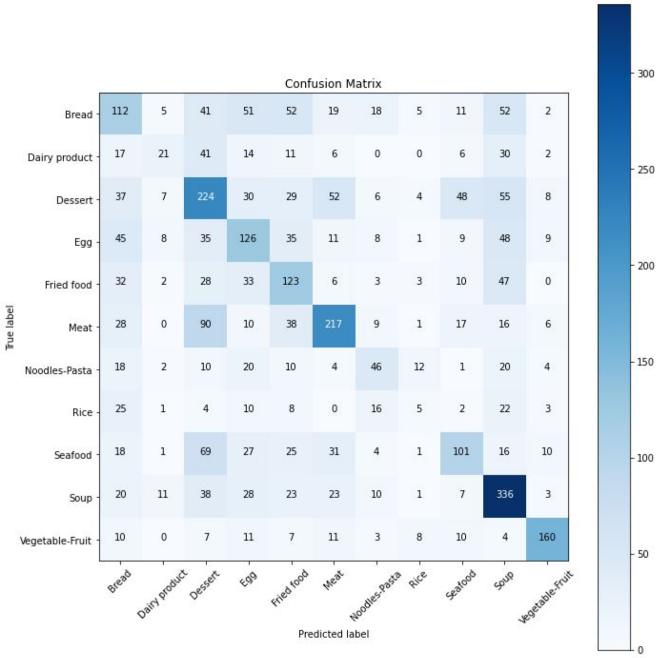


Fig. 2. Sample of the processed dataset for TL45

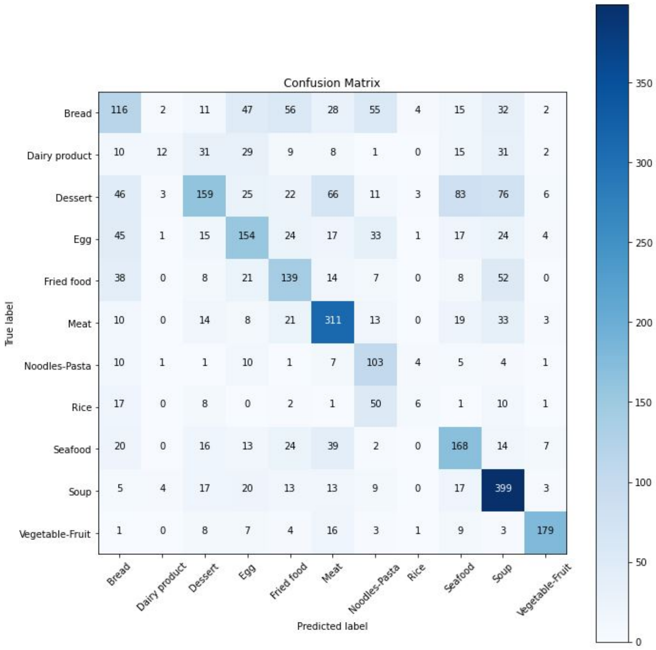


	Bread	Dairy product	Dessert	Egg	Fried food	Meat	Noodles-Pasta	Rice	Seafood	Soup	Vegetable-Fruit
Bread	112	5	41	51	52	19	18	5	11	52	2
Dairy product	17	21	41	14	11	6	0	0	6	30	2
Dessert	37	7	224	30	29	52	6	4	48	55	8
Egg	45	8	35	126	35	11	8	1	9	48	9
Fried food	32	2	28	33	123	6	3	3	10	47	0
Meat	28	0	90	10	38	217	9	1	17	16	6
Noodles-Pasta	18	2	10	20	10	4	46	12	1	20	4
Rice	25	1	4	10	8	0	16	5	2	22	3
Seafood	18	1	69	27	25	31	4	1	101	16	10
Soup	20	11	38	28	23	23	10	1	7	336	3
Vegetable-Fruit	10	0	7	11	7	11	3	8	10	4	160

Fig. 3. CNN1 confusion matrix for one batch of test data

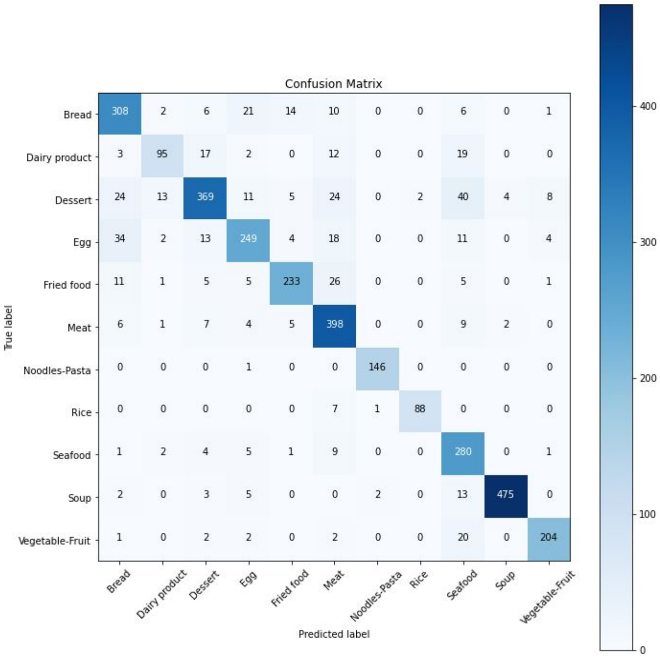
education. This free tool provided by Google eliminates the need for setup and allows for the execution of high-performance GPU codes. Python programming language was utilized for the experiment, while TensorFlow and Keras libraries were used for the DL codes. The dataset consisted of 16,000 food images, divided into 11 classes. To split the dataset into train, validation, and test sets, the Split folder function in Python was employed. The pre-processing method outlined was followed to fulfill the various requirements of the CNNs and pre-trained model.

The results of the training and validation loss and accuracy are depicted in Fig.6. The training loss curve in the figure exhibits a gradual decrease, indicating that the model is learning and improving its performance on the training data.



	Bread	Dairy product	Dessert	Egg	Fried food	Meat	Noodles-Pasta	Rice	Seafood	Soup	Vegetable-Fruit
Bread	116	2	11	47	56	28	55	4	15	32	2
Dairy product	10	12	31	29	9	8	1	0	15	31	2
Dessert	46	3	159	25	22	66	11	3	83	76	6
Egg	45	1	15	154	24	17	33	1	17	24	4
Fried food	38	0	8	21	139	14	7	0	8	52	0
Meat	10	0	14	8	21	311	13	0	19	33	3
Noodles-Pasta	10	1	1	10	1	7	103	4	5	4	1
Rice	17	0	8	0	2	1	50	6	1	10	1
Seafood	20	0	16	13	24	39	2	0	168	14	7
Soup	5	4	17	20	13	13	9	0	17	399	3
Vegetable-Fruit	1	0	8	7	4	16	3	1	9	3	179

Fig. 4. CNN20 confusion matrix for one batch of test data



	Bread	Dairy product	Dessert	Egg	Fried food	Meat	Noodles-Pasta	Rice	Seafood	Soup	Vegetable-Fruit
Bread	308	2	6	21	14	10	0	0	6	0	1
Dairy product	3	95	17	2	0	12	0	0	19	0	0
Dessert	24	13	369	11	5	24	0	2	40	4	8
Egg	34	2	13	249	4	18	0	0	11	0	4
Fried food	11	1	5	5	233	26	0	0	5	0	1
Meat	6	1	7	4	5	398	0	0	9	2	0
Noodles-Pasta	0	0	0	1	0	0	146	0	0	0	0
Rice	0	0	0	0	0	7	1	88	0	0	0
Seafood	1	2	4	5	1	9	0	0	280	0	1
Soup	2	0	3	5	0	0	2	0	13	475	0
Vegetable-Fruit	1	0	2	2	0	2	0	0	20	0	204

Fig. 5. TL45 confusion matrix for one batch of test data

On the other hand, the validation loss curve drops slightly but then starts to rise again. This behavior suggests that the model is not generalizing well to new data and is likely overfitting to the training data.

The reason for the overfitting is that the number of parameters that the model can learn is too high compared to the features in our image data. This means that the model is trying to fit the noise or the idiosyncrasies of the training data instead of capturing the underlying patterns that are common to both the training and validation data. Therefore, the model is not able

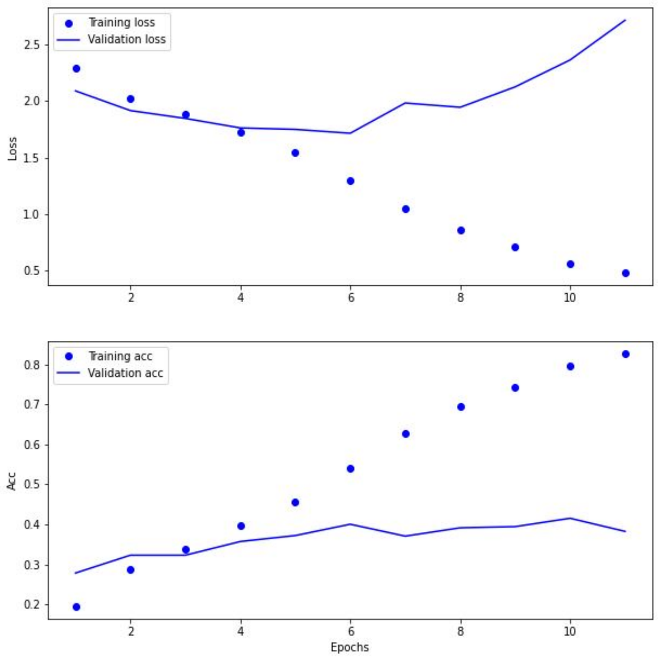


Fig. 6. validation/training loss and accuracy for CNN1

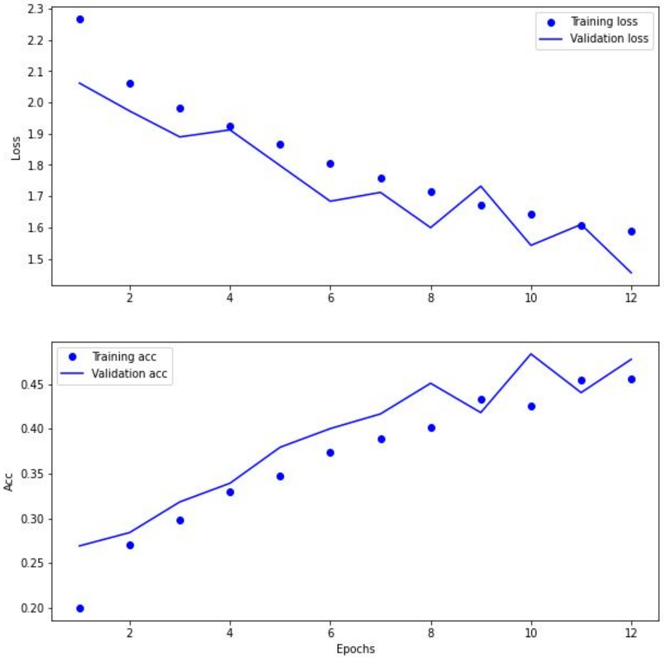


Fig. 7. validation/training loss and accuracy for CNN20

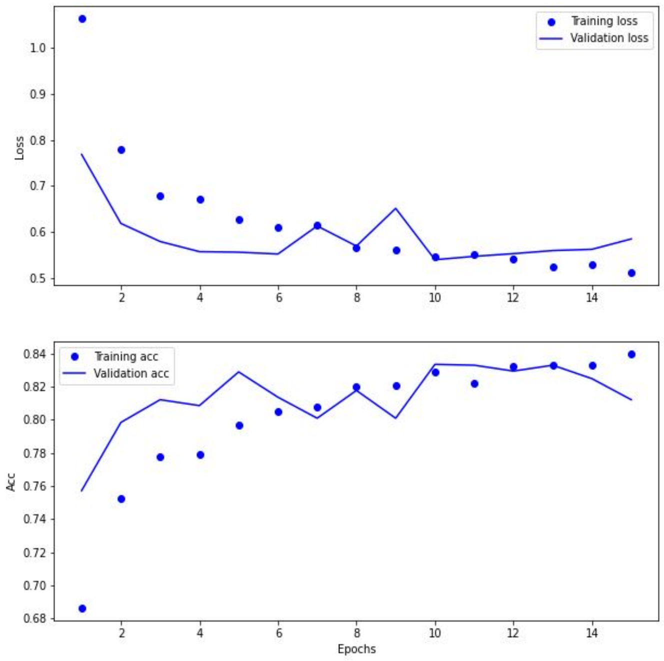


Fig. 8. validation/training loss and accuracy for TL45

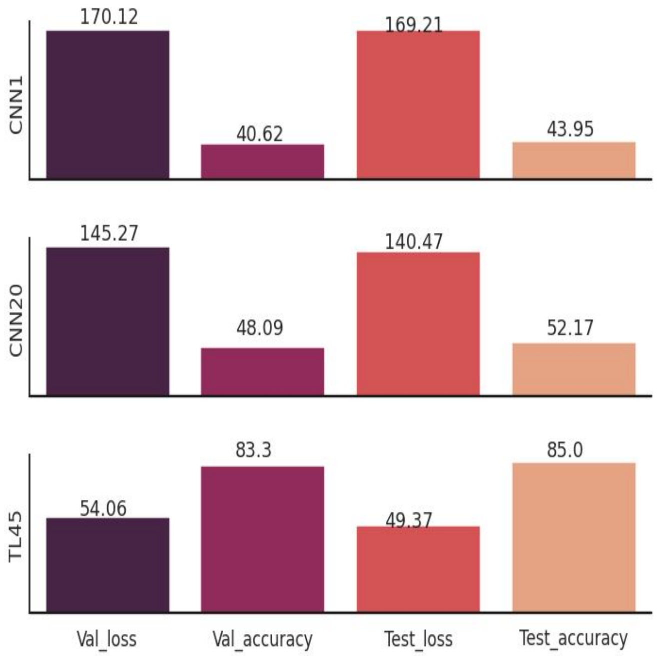


Fig. 9. Comparison of validation/test loss and accuracy for three models CNN1, CNN20, and TL45

to perform well on new or unseen data.

Fig.7 demonstrates that the performance of the CNN20 model is superior to that of the CNN1 model. One possible explanation for this is that the CNN20 model does not suffer from overfitting, which is a common problem in deep learning models. This is achieved through the use of data augmentation techniques, which help to expand the size and diversity of the training data.

Although the validation accuracy of the CNN20 model is still not considered high enough, reaching a maximum of 48.09%, it is reasonable given the limited number of epochs that the model has undergone. The output patterns indicate that the model could potentially learn more if it was trained for a longer period, as evidenced by the fact that the accuracy has not plateaued.

The graph displayed in Fig.8 illustrates the loss and accuracy metrics for the InceptionResNetV2 model (TL45). Unlike the CNN1 and CNN20 approaches, this model was pre-trained on the ImageNet dataset and utilizes its own data pre-processing techniques and training weights.

The model underwent a training process of 15 epochs, during which the feature extractors of all models were frozen while the classifiers were trained using our training and validation data, along with pre-trained weights.

The results indicate that the TL45 model outperforms both the CNN1 and CNN20 models. Specifically, the validation accuracy achieved by the TL45 model reached 83.3%, which is significantly higher than that of the other models. The gap between the validation accuracy and training loss is also relatively small and reasonable, which suggests that the model is not overfitting to the training data.

Fig.9 provides a visual comparison of the three models studied in terms of validation/test loss and accuracy. The results show that the TL45 model outperformed the other models, achieving an accuracy of 85% in classifying unseen test data.

Additionally, the CNN20 model performed better than the CNN1 model. This is likely due to the use of data augmentation techniques in the CNN20 model, which helps to avoid overfitting by increasing the diversity and size of the training data.

Overall, the experimental results indicate that the TL45 model is the most effective model in terms of accuracy, while the CNN20 model performs better than the CNN1 model. These findings suggest that the use of pre-trained models, such as the TL45 model, and data augmentation techniques, such as those employed by the CNN20 model, can improve the accuracy of image classification models.

V. CONCLUSION

In this study, a basic TL model was investigated for categorizing food and then evaluated its performance against two other CNN models. One utilized data augmentation techniques. The experiment was conducted with a limited number of training iterations. The results showed that the TL-based method performed better than the other models. This was not surprising given that the dataset used in the study was small. Furthermore, it can be acknowledged that data augmentation could potentially improve the performance of the models.

REFERENCES

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

[2] K. Sukvichai, P. Maolanon, K. sawanyawat and W. Muknumporn, "Food categories classification and Ingredients estimation using CNNs on Raspberry Pi 3," 2019 10th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES), Bangkok, Thailand, 2019, pp. 1-6, doi: 10.1109/ICTEmSys.2019.8695967.

[3] K. T. Islam, S. Wijewickrema, M. Pervez and S. O'Leary, "An Exploration of Deep Transfer Learning for Food Image Classification," 2018 Digital Image Computing: Techniques and Applications (DICTA), Canberra, ACT, Australia, 2018, pp. 1-5, doi: 10.1109/DICTA.2018.8615812.

[4] M. F. Ahamed, O. Sarkar and A. Matin, "Instance Segmentation of Visible Cloud Images Based on Mask R-CNN Applying Transfer Learning Approach," 2020 2nd International Conference on Advanced Information and Communication Technology (ICAICT), Dhaka, Bangladesh, 2020, pp. 257-262, doi: 10.1109/ICAICT51780.2020.9333531.

[5] S. Zakzouk, A. Saafan, M. -A. Sayed, M. A. Elattar and M. Saeed Darweesh, "Light-Weight Food Image Classification For Egyptian Cuisine," 2022 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), Sakheer, Bahrain, 2022, pp. 581-586, doi: 10.1109/3ICT56508.2022.9990862.

[6] A. ÅđengÃijr, Y. Akbulut and ÃIJ. Budak, "Food Image Classification with Deep Features," 2019 International Artificial Intelligence and Data Processing Symposium (IDAP), Malatya, Turkey, 2019, pp. 1-6, doi: 10.1109/IDAP.2019.8875946.

[7] Y. Liqiang, M. Erdt and W. Lipo, "Adaptive Transfer Learning To Enhance Domain Transfer In Brain Tumor Segmentation," 2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI), Nice, France, 2021, pp. 1873-1877, doi: 10.1109/ISBI48211.2021.9434100.

[8] T. -W. Li and G. -C. Lee, "Performance Analysis of Fine-tune Transferred Deep Learning," 2021 IEEE 3rd Eurasia Conference on IOT, Communication and Engineering (ECICE), Yunlin, Taiwan, 2021, pp. 315-319, doi: 10.1109/ECICE52819.2021.9645649.

[9] N. An, "Xception Network for Weather Image Recognition Based on Transfer Learning," 2022 International Conference on Machine Learning and Intelligent Systems Engineering (MLISE), Guangzhou, China, 2022, pp. 330-333, doi: 10.1109/MLISE57402.2022.00072.

[10] L. Li and W. Seo, "Deep Learning and Transfer Learning for Skin Cancer Segmentation and Classification," 2021 IEEE 21st International Conference on Bioinformatics and Bioengineering (BIBE), Kragujevac, Serbia, 2021, pp. 1-5, doi: 10.1109/BIBE52308.2021.9635175.

[11] N. Jethwa, H. Gabajiwala, A. Mishra, P. Joshi and P. Natu, "Comparative Analysis between InceptionResnetV2 and InceptionV3 for Attention based Image Captioning," 2021 2nd Global Conference for Advancement in Technology (GCAT), Bangalore, India, 2021, pp. 1-6, doi: 10.1109/GCAT52182.2021.9587514.

[12] M. Naveenkumar, S. Srithar, B. Rajesh Kumar, S. Alagumuthukrishnan and P. Baskaran, "InceptionResNetV2 for Plant Leaf Disease Classification," 2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 2021, pp. 1161-1167, doi: 10.1109/I-SMAC52330.2021.9641025.