Düzce University
Journal of Science & Technology

Research Article

Deep Learning with Limited Data: Advanced Classification Approaches Through Few-Shot Learning and Prototype Networks

Sara ALTUN GÜVEN [a,*], Buket TOPTAŞ[b]

[a] *Department of Computer Eng, Faculty of Engineering, Tarsus University, Tarsus/Mersin, TURKEY*
[b] *Software Eng. Dept., Engineering and Natural Science Faculty, Bandırma Onyedi Eylül University, Bandırma/Balıkesir, TURKEY*
** Corresponding author's e-mail address: saraguven@tarsus.edu.tr*
DOI: 10.29130/dubited.1460641

ABSTRACT

Classification problems in the fields of machine learning and artificial intelligence facilitate the extraction of meaningful information from data by assigning inputs to specific categories. Classification processes offer solutions for a wide range of areas, including health, agriculture, education, and sports. However, the classification process typically requires a large amount of labeled data. Accessing a large volume of labeled data is costly and time-consuming. The few-shot learning method has been utilized to address this issue, allowing models to learn new tasks with minimal examples. In this article, pre-trained deep network architectures have been fed into prototype networks, creating representative examples for each class. Thus, the category to which new data belongs is determined based on its similarity to the prototypes. Experimental studies have been conducted on the Food101 and Oxford-III Pet datasets, and the experimental results have been measured using four different evaluation metrics. The results have been presented and interpreted both in table form and graphically. In comparing classification accuracy, the metrics of Accuracy, F1_Score, Precision, and Recall were utilized. For the Oxford-III Pet dataset, ResNet18 demonstrated the best classification performance with metric values of 0.9986, 1, 1, and 1 for Accuracy, F1_Score, Precision, and Recall, respectively. In the case of the Food101 dataset, EfficientNetB0 achieved the highest classification performance, with values of 0.9320, 0.93, 0.94, and 0.93 for Accuracy, F1_Score, Precision, and Recall, respectively.

Keywords: *Classification, few-shot learning, transfer learning*

Sınırlı Veri ile Derin Öğrenme: Birkaç Atışlı Öğrenme ve Prototip Ağlar Aracılığıyla Gelişmiş Sınıflandırma Yaklaşımları

ÖZ

Sınıflandırma problemleri, makine öğrenimi ve yapay zekâ alanında, girdileri belirli kategorilere atayarak verilerden anlamlı bilgi çıkarılmasını sağlar. Sınıflandırma işlemleri; sağlık, tarım, eğitim ve spor gibi geniş bir alan için çözümler sunar. Ancak, sınıflandırma işlemi yapılırken genellikle büyük miktarda etiketli veriye ihtiyaç duyulur. Büyük miktarda etiketli veriye ulaşmak maliyetli ve zaman alıcıdır. Bu problemin çözebilmek için birkaç atışlı öğrenme yöntemi ile modelin çok sınırlı örneklerle yeni görevleri öğrenmesine olanak tanınmıştır. Bu makalede, önceden eğitilmiş derin ağ mimarileri prototip ağlara beslenmiş ve her sınıf için temsilci örnekler oluşturulmuştur. Böylece, yeni verilerin hangi kategoriye ait olduğu prototiplere olan benzerliğe göre belirlenmiştir. Deneysel çalışmalar, Food101 ve Oxford-III Pet veri setleri üzerinde denenmiş ve deneysel sonuçlar dört farklı değerlendirme metriği ile ölçülmüştür. Deneysel sonuçlar hem tablo olarak hem de grafiksel olarak gösterilmiş ve yorumlanmıştır. Sınıflandırma doğruluğunu karşılaştırmak için Doğruluk, F1_Skoru, Kesinlik ve Duyarlılık metrikleri kullanılmıştır. Oxford-III Pet veri seti için, ResNet18 mimarisi sırasıyla Doğruluk, F1_Skoru, Kesinlik ve Duyarlılık için 0.9986, 1, 1 ve 1 değerleriyle en iyi sınıflandırma performansını göstermiştir. Food101

Received: 28/03/2024, Revised: 15/04/2024, Accepted: 20/12/2024

veri seti için ise EfficientNetB0 mimarisi sırasıyla 0.9320, 0.93, 0.94 ve 0.93 değerleriyle Doğruluk, F1_Skoru, Kesinlik ve Duyarlılık açısından en yüksek sınıflandırma performansına ulaşmıştır.

Anahtar Kelimeler: *Sınıflandırma, az atışlı öğrenme, transfer öğrenme*

I. INTRODUCTION

Classification is an essential procedure in data science that divides data into predetermined categories. Sorting a given collection of input samples into preset categories is the aim of classification. Classification processes, as applied in various fields such as glaucoma detection using fundus images [1], network traffic classification to distinguish traffic [2], and atrial fibrillation detection from electrocardiogram recordings [3], are critically important because they automate decision-making processes across different field.

The automatic classification of data has been accomplished using machine learning techniques known as supervised and unsupervised learning methods [3-5]. However, the emergence of deep neural network architectures has accelerated classification tasks, producing successful outcomes in this area. In the realm of classification, Convolutional Neural Network (CNN) architectures like AlexNet [6], GoogLeNet [7], VggNet [8], ResNet [9], and DenseNet [10] have had a tremendous influence and achieved notable accomplishments. While the ability of deep network architectures to produce successful results and operate quickly is viewed as advantageous, these methods also have their limitations. Deep network architectures require substantial hardware resources and a large amount of data. To address the hardware dependency issue, servers such as GoogleColab and Amazon AWS have been made available. However, the need for large data remains a significant constraint. Researchers have developed synthetic data augmentation methods, such as generative adversarial networks, to meet the demand for data augmentation; however, challenges remain regarding the reliability and dependency on large datasets. Even when a substantial amount of data is available, finding and labeling labeled data is quite burdensome. This process is particularly costly in terms of time with multi-class data. The scarcity of data leads to problems with models being unable to learn, which adversely affects model performance.

In 2006, Fei-Fei Li et al. [11] proposed a method known as "One-Shot Learning", paving the way for the operation of deep neural network architectures with limited data. Subsequently, the development of the concept of "meta-learning" has facilitated the advancement of methods referred to as "Few-Shot Learning (FSL)". FSL has been utilized in numerous areas including classification, object detection, and segmentation. Paeedeh et al. [12] proposed an "Adaptive Transformer Network" using few-shot learning. The aim of the method is to detect domain shifts between the base task and the target task. Zhao et al. [13] proposed a self-attention mechanism-based FSL. The objective here is to transform the features obtained via the transfer network and to expand the support set with query samples that have high reliability. Snell et al. [14] designed "Prototypical Networks (ProtoNet)" for the few-shot classification problem. This method generates state-of-the-art results effectively and simply, even without the complex extensions developed for matching networks among meta-learning methods. In brief, due to its simplicity and effectiveness, this network structure is considered a promising approach for few-shot learning. Sung et al. [15] proposed the "Relation Network" method. This network undergoes end-to-end training from scratch. Extensive experiments conducted across five different benchmarks demonstrate that it is a unified and effective approach for both zero-shot learning and few-shot learning. Wang et al. [16] introduced the "Simple Shot" method. This method investigates the accuracy of nearest neighbor baselines without the need for meta-learning. As a result, it has been observed that simple feature transformations are sufficient to achieve competitive few-shot learning accuracies. Gülcü and Alkan [17] have examined the Model-Agnostic Meta-Learning (MAML) and ProtoNet algorithms for the few-shot learning problem. This study determined that the MAML algorithm yields better results than ProtoNet with fewer examples; however, ProtoNet is able to generalize better when there are more examples. Işık [18] explores the use of few-shot learning algorithms to improve classification performance in scenarios where traditional deep learning methods fail due to a lack of training data. Experimental results were obtained by classifying tomato diseases in the PlantVillage dataset. This approach suggests the potential inclusion of attention mechanisms in feature extraction processes and proposes new areas of research within few-shot learning methodologies. Argüse et al. [19] have introduced FSL algorithms for plant leaf classification with small datasets using deep learning. The PlantVillage dataset was utilized in this study. FSL was benchmarked using Siamese networks and Triplet loss. Consequently, it has been observed that learning new plant leaf and disease types with very small datasets is feasible using deep learning with Siamese networks and Triplet loss. Wang et al. [20] proposed a few-shot learning model based on the Siamese network for classifying plant leaves. This metric also utilizes a k-nearest neighbors classifier. The Flavia, Swedish, and Leafsnap datasets were employed to evaluate the method. Experimental results demonstrate that the proposed method can achieve high classification accuracy with a small supervised sample size. Frikha et al. [21] focus on the few-shot and One-Class Classification (OCC) problem. This method aims to learn a model particularly suitable for the few-shot one-class classification process. The method was evaluated across 8 datasets. Consequently, it has been experimentally observed that the proposed data sampling technique enhances the performance of newer meta-learning algorithms in few-shot OCC scenarios, delivering state-of-the-art results for this problem. Chen et al. [22] have proposed the joint use of a self-supervised learning approach with an embedding network for few-shot image classification. Studies conducted on four datasets have proven that the proposed method can achieve state-of-the-art results. Krenzer et al. [23] utilized deep learning architectures combined with a few-shot learning approach to automate the classification of polyps. They developed classification methods based on polyp shape and texture-surface patterns. The classification method based on texture-level patterns is termed NICE. This few-shot learning-based NICE classification achieved an accuracy of 81.13% when applied to a limited dataset. Liu et al. [24] have proposed a deep few-shot learning method for hyperspectral image classification. The aim is to facilitate hyperspectral image classification with less data. The widely used HSI dataset was employed for performance evaluation. Consequently, it has been demonstrated that the method can achieve better classification accuracy with just a few labeled examples. Kang and Cho [25] have proposed an integrated few-shot learning method for classification and segmentation tasks. Experimental results have demonstrated that the proposed method process exhibits promising performance and achieves state-of-the-art results on standard few-shot segmentation benchmarks. Hu et al. [26] have proposed a transfer-based few-shot learning method. This approach aims to preprocess feature vectors to approximate them to Gaussian-like distributions and utilize an algorithm based on this preprocessing outcome. The results have been evaluated on benchmarks. The method has provided accuracy in both 1-shot and 5-shot classifications and has been observed to yield significant outcomes with a minimal number of hyper parameters. Kim et al. [27] proposed an Edge-Labeling Graph Neural Network (EGNN) for few-shot learning. This method adapts the deep neural network on an edge-labeling graph and executes this adaptation iteratively. Experimental results have shown that the proposed EGNN outperforms other few-shot learning algorithms in both supervised and semi-supervised few-shot image classification tasks.

This article analyzes the comparison of FSL systems inspired by transfer learning on prototypical networks. Table 1 illustrates the general progression of the article.

Table 1. *Overview of the Article Analysis*

Problem	Method	Challenge
Limited labeled data	Prototypical networks	Can limit their flexibility in highly variable datasets.
Diverse image categories	Pre-trained CNNs	Computational cost varies by model
Few-shot recognition	5-shot-5-way FSL setup	Limited scalability to larger datasets

Here, the performance of FSL is improved by applying learned information about data classes, such as labeled data for multiple classes from large datasets, to new classes. This approach provides a thorough analysis of how well CNN architectures pre-trained on two distinct datasets inside Prototypical Networks perform.

The remainder of the paper is organized as follows. Section II contains the Materials and Methods. This section discusses the datasets used, the methodologies employed, and the implementation environment. Section III is the Experimental Results and Discussion, where the outcomes of the experimental study are presented and discussed. The final section concludes the paper.

II. MATERIAL AND METHODS

A. USING DATASETS

In this study, the publicly available Oxford-III Pet [28] and Food101 [29] classification datasets were used to obtain the experimental results.

A. 1. Oxford-III Pet Dataset

The Oxford-III Pet dataset [28] focuses on pet animals. This dataset comprises 37 different categories, with nearly 200 images in each category [28]. Each image includes the breed (species), pixel-level trimap segmentation, and the head ROI (Region of Interest). The breed label identifies the pet's species and is used in classification tasks. The head ROI denotes the specified region of the pet's head within the image. Pixel-level trimap segmentation is a labeling process that indicates the likelihood of each pixel belonging to a specific area or class. This is a crucial component in clarifying the boundaries of pets in images. The diversity in images and ground truth labels also provide a rich resource for the development of deep learning and artificial intelligence techniques. Figure 1 showcases examples from the Oxford-III Pet dataset [28].

Figure 1. *Examples of images from the Oxford-III Pet dataset*

A. 2. Food101 Dataset

The Food101 dataset [29] encompasses 101 food classification categories, consisting of 101,000 images; each class contains a total of 1,000 images. Within each class, there are 250 manually curated test images and 750 training images. The training images have been intentionally left uncleaned (to facilitate better training) and contain a minimal amount of noise [29]. All images have been resized to a maximum of 512×512 pixels. Figure 2 showcases examples from the Food101 dataset [29].

Figure 2. *Examples of images from the Food101 dataset*

B. METHOD

In this section, a transfer learning framework for few-shot learning is presented. The proposed framework consists of three primary steps. Initially, features are extracted from base class data using three principal pre-trained deep network architectures. In the second step, the feature extractor is employed to derive features from new class data, which are then provided to the prototypical network as support and query sets. The final step involves measuring the classification success of the model. This article utilizes two large-scale datasets containing base classes. The task of FSL is to solve the N-way-K-shot problem for each class within the dataset. The flow-chart diagram of the method is presented in Figure 3.

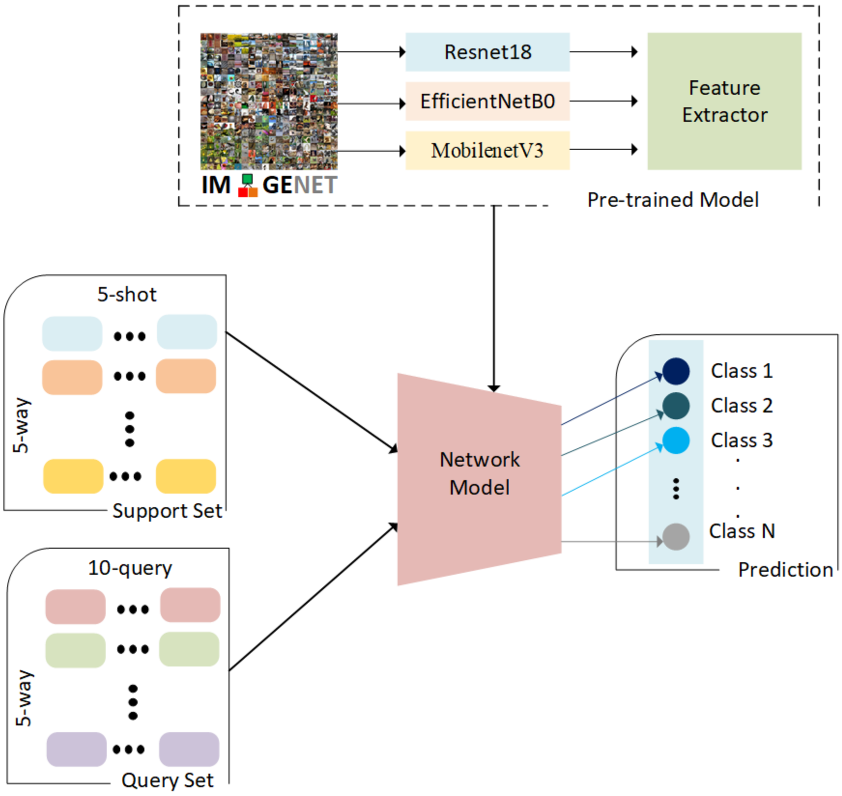

Figure 3. *Flow-Chart Diagram*

C. FEW SHOT LEARNING

Few-shot learning is a learning method that addresses the limited data problem. In this approach, the model aims to quickly adapt to new environments with extremely few examples. FSL is a significant research area for classification, object recognition, and other tasks. It operates over base and novel class sets. The base class is the one with a large number of labels. The novel class is the part with very few training examples. Here, the objective is to enable the recognition of new classes with even a minimal number of examples. The model is trained on the base classes and extracts general features from the examples it sees. Then, the model is tested on the novel class. Here, for each novel class, the model receives a few "support" examples and a set of "query" examples. Support consists of the few examples used for training. Query includes the test examples that the model needs to classify. ProtoNets are utilized to overcome the overfitting problem caused by the scarcity of data in labeled classes. ProtoNets extract features from a few support examples of each new class and calculate the average of all examples within a class, thereby selecting a representative feature for each class. These selected representative features are termed as "prototypes." Hence, acting as a representative for the respective category, they rapidly classify new examples and provide generalization. Prototype networks have proven their success in few-shot classification by achieving high performance, demonstrating their effectiveness [30-32]. Figure 4 represents a visual for prototype networks. In Figure 4, an embedding space and the points within this space are shown. The embedding space is a mathematical space used to obtain a more manageable, typically lower-dimensional representation of the data. The colored regions in the figure represent data examples belonging to a specific class, and at the center of each class, the "prototype" of that class is located. During the learning process, the model uses these prototypes to classify new examples. A point (i.e., a test example) in the embedding space is labeled as belonging to the class whose region it falls into.

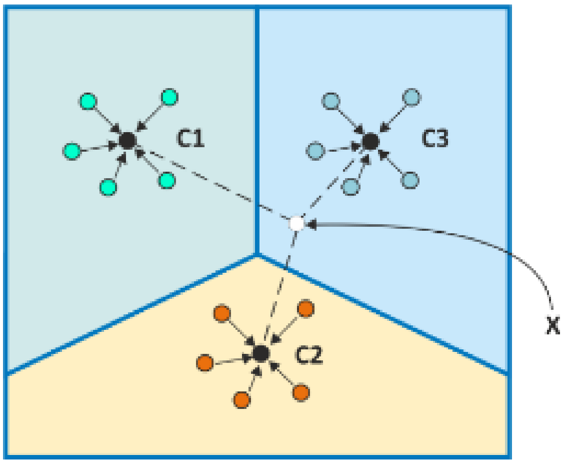

Figure 4. *Prototypical network [14]*

C. TRANSFER LEARNING

Transfer learning is the reuse of a model trained for one task for another task. It occurs in two stages. In the first stage, there is an original dataset and a task for which this dataset will be applied. In the second stage, there is a secondary dataset and a new task for which this dataset will be applied. If the data transfer is made between similar domains, it is classified as homogeneous learning; if made between different domains, it is classified as heterogeneous learning. The advantage of transfer learning is that it reduces overfitting and improves the model's generalization. In deep network architectures, the training of models is accelerated and faster convergence is achieved in new tasks by using previously trained network architectures. The ImageNet dataset [33], which contains more than 1.2 million natural images, includes pre-trained CNN architectures. The transfer learning approaches used in this article are the

ResNet18 [9], EfficientNetB0 [34], and MobileNetV3 [35] architectures. Detailed information about these trained architectures is presented in Table 2. The ResNet18 [9] architecture has 18 layers. In deep network architectures, as the depth increases, the vanishing gradient problem emerges. It has been proposed to overcome this problem. This architecture consists of structures called "Residual Blocks." In traditional deep network architectures, each layer is given as input to the next layer. However, if there are residual blocks, each layer feeds not only the next layer but also distant layers through skip connections. The EfficientNetB0 [34] architecture, unlike other traditional deep network architectures, uses the "Compound Scaling" method. This method finds the optimal structure for the model by scaling the resolution, depth, and width dimensions together. The architecture also efficiently reduces the number of parameters using blocks called Mobile Inverted Bottleneck Convolution (MBConv). The MobileNetV3[35] architecture has a lightweight network structure, making it preferable for hardware with limited computational power. Unlike traditional deep network structures, it utilizes depth wise separable convolutions (DSC), squeeze-and-excitation (SE) blocks, and inverted residual blocks. DSC performs the convolution operation by dividing it into depth and point operations. SE emphasizes important features by modeling the relationships between channels, focusing on the channel dimension and disregarding the spatial dimension of the target information. Inverted residual blocks are used to connect input and output features to the same channel, thus preventing excessive memory consumption.

Table 2. *Deep learning architectures used transfer learning*.

Model	**Size [MB]**	**Number of parameters**	**Depth**
ResNet18 [9]	45	11.7M	18
EfficientNetB0 [34]	20.5	5.3M	224
MobileNetV3 [35]	9.8	2.5M	14

D. IMPLEMENTATION DETAILS

In this study, the Prototypical Network from FSL network structures was utilized. The implementation was carried out in Python, leveraging the Torch library and the easyfsl library designed for few-shot learning. A Flatten layer was used as the convolutional network layer. Table 3 displays the fixed parameters determined for FSL. Table 4 provides the training details of the deep network architectures used for transfer learning.

Table 3. *Few-Shot Learning Constants for the Food101 and Oxford-III Pet Datasets*

Parameter(s)	**Value(s)**
Way	5
Shot	5
Query	10
Evaluation Task	100

Table 4. *Training Information for the Food101 and Oxford-III Pet Datasets*

Parameter(s)	**Value(s)**
Training Episodes	60000
Validation task	100
Optimizer	Adam
Learning rate	---
Criterion	Cross entropy loss

E. EVALUATION METRIC

Evaluation metrics assess the performance of conducted experimental studies, allowing for the comparison of methods and determination of their success rates. In this article, the four most commonly used evaluation metrics have been employed. These metrics are *accuracy*, *precision*, *recall*, and *F1_Score*. *Accuracy* (*Acc*) measures the proportion of correctly predicted examples within the total samples, serving as the most fundamental metric in classification problems. It gauges the model's ability to correctly classify all classes. *Precision* (*Pr*) represents the ratio of positive examples correctly classified by the model to all examples classified as positive. *Recall* (*R*) denotes the ratio of true positive examples to the sum of true positive and false negative examples identified by the model, aiming to reduce false negatives. The *F1_Score*, or *Dice*, is the harmonic mean of precision and recall, indicating the balance between them. The mathematical expressions for these evaluation metrics are provided sequentially from Equation (1) to (4).

$$Acc = \frac{Correct\ Predictions}{Total\ Predictions} \quad (1)$$

$$Pr = \frac{True\ Positives}{True\ Positives + False\ Positives} \quad (2)$$

$$R = \frac{True\ Positives}{True\ Positives + False\ Negative} \quad (3)$$

$$F1\_Score = 2 \times \left(\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}\right) \quad (4)$$

III. EXPERIMENTAL RESULT and DISCUSSING

The experimental results of the classification study performed with FSL using transfer learning are presented in this section. Each CNN was tested over two datasets with 60,000 iterations, and the loss function was recorded at every epoch. Figure 5 displays the loss functions during the training phase. According to these graphs, the value of the loss showed a rapid decline at the beginning of the training. This can be interpreted as an indicator that the model began learning quickly. A general look at the graphs indicates that after approximately 10,000 training segments, there is a deceleration in the rate of decrease of the loss value for every CNN architecture. This has been interpreted as the model reaching a sort of saturation in its learning process, learning less with each iteration. In later iterations, the loss value showed less decline and slight fluctuations appeared in the graph. These fluctuations occurred as the model was learning the features within the training set. Towards the end, the loss value stabilizes, indicating that the model has reached saturation in its training process, and further training does not significantly affect the loss value. In general, this suggests that the model has achieved a certain level of learning on the training set and offers insights into how it will perform on test data.

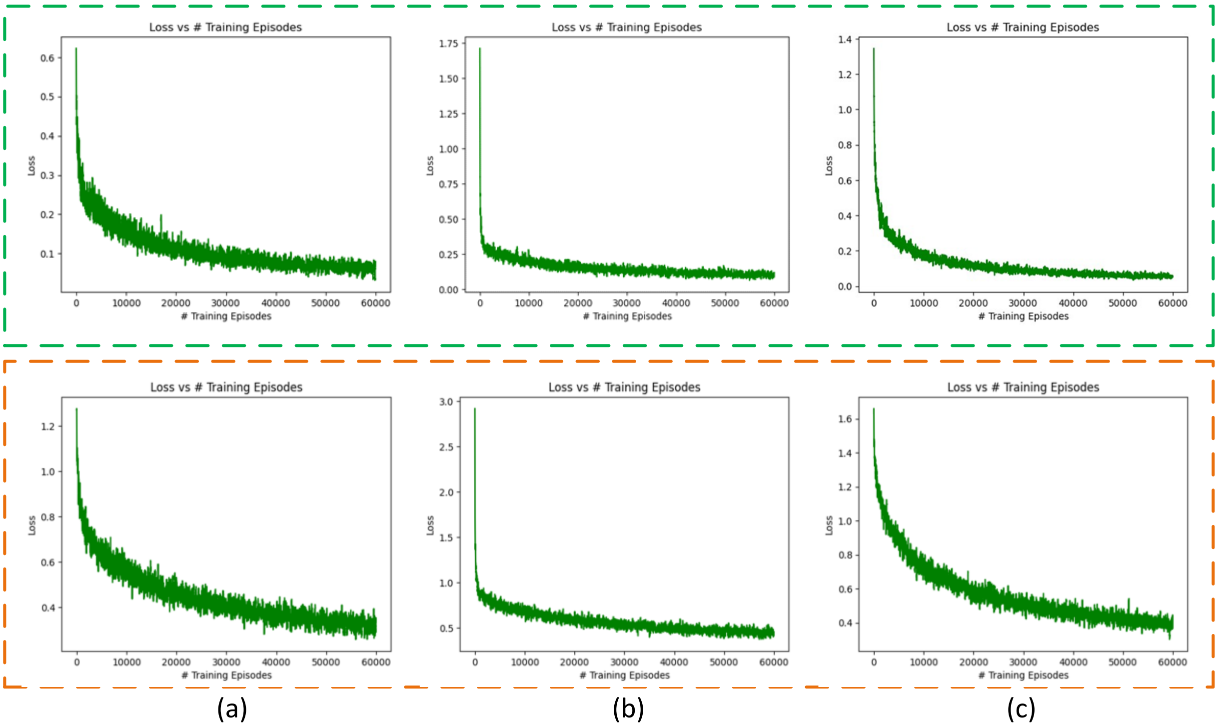

Figure 5. *Training Loss. Green frames are related to the Oxford-III Pet dataset; Red frames correspond to the Food101 dataset. (a) Result of the EfficientNetB0 [34] architecture, (b) Result of the MobileNetV3* [35] *architecture, (c) Result of the ResNet18* [9] *architecture.*

Tables 5 and 6 display the test results of the model within two datasets. According to this, the Oxford-III Pet dataset has achieved better classification success compared to the Food101 dataset. The best-performing architecture for both datasets was the EfficientNetB0 [34] architecture (in bold font). The second-best performance was shown by the ResNet18 [9] architecture (italicized and underlined font). The *F1_Score*, *Precision*, and *Recall* values have provided both high and balanced results. This indicates that the model accurately predicts both positive and negative examples in a balanced manner.

Table 5*. Accuracy Comparison of Network Models for the Oxford-III Pet Dataset*

	Acc.	**F1_Score**	**Pr**	**R**
EfficientNetB0 [34]	*0.9974*	*1*	*1*	*1*
MobileNetV3 [35]	0.9936	0,99	0,99	0,99
ResNet18 [9]	**0.9986**	**1**	**1**	**1**

Table 6. *Accuracy Comparison of Network Models for the Food101 Dataset*

	Acc	**F1_Score**	**Pr**	**R**
EfficientNetB0 [34]	**0.9320**	**0.93**	**0.94**	**0.93**
MobileNetV3 [35]	0.9092	0.91	0.92	0.91
ResNet18 [9]	*0.9176*	*0.92*	*0.93*	*0.92*

IV. CONCLUSION

In this article, the performance outcomes of pre-trained network architectures for recognizing food and pets in a few-shot learning context have been investigated. A prototypical network was utilized as the FSL architecture. Pre-trained network architectures such as EfficientNetB0 [34], ResNet18 [9], and MobileNetV3 [35] were employed. All experimental work was conducted over 60,000 iterations in a 5-shot-5-way configuration. The study observed that pre-trained network architectures like EfficientNetB0 [34], ResNet18 [9], and MobileNetV3 [35] are quite successful in few-shot recognition of food and pets.

Particularly, EfficientNetB0 [34], offering the highest accuracy rate, has proven to be a highly suitable option for these types of tasks. ResNet18 [9] and MobileNetV3 [35] also achieved competitive results within their capabilities and, depending on application requirements and hardware limitations, may be preferred for their lower computational cost and speed. The experimental studies conducted over 60,000 iterations under a 5-shot-5-way setup have demonstrated the robustness and generalization capability of these architectures in the context of FSL. Future studies might explore hyper parameter tuning, different feature fusion methods, and the impact of various data augmentation techniques to further enhance the model.

Declaration of Ethical Standards
The authors remained faithful to all ethical rules

Credit Authorship Contribution Statement
Authors 1: Software, Investigation, Writing – original draft, Methodology,
Authors 2: Writing – review and editing, Validation, Conceptualization,

Declaration of Competing Interest
There is no conflict of interest between authors

V. REFERENCES

[1] Toptas, B., Hanbay, D. "The Separation of Glaucoma and Non-Glaucoma Fundus Images using EfficientNet-B0." *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi*, vol. 11, No. 4, pp. 1084–1092, 2022.

[2] Gudla, R., Vollala, S., Srinivasa, K. G., Amin, R. "A novel approach for classification of Tor and non-Tor traffic using efficient feature selection methods." *Expert Systems with Applications*, Article ID 123544, 2024.

[3] Gündüz, A. F., Talu, M. F. "Atrial fibrillation classification and detection from ECG recordings." *Biomedical Signal Processing and Control*, vol. 82, Article ID 104531, 2023.

[4] Toptaş, B., Hanbay, D. "Retinal blood vessel segmentation using pixel-based feature vector." *Biomedical Signal Processing and Control*, vol. 70, Article ID 103053, 2021.

[5] Toptaş, M. *Orman yangınlarının görüntü işleme yöntemleri ile tespit edilmesi ve sınıflandırılması* (Yüksek Lisans tezi, İnönü Üniversitesi Fen Bilimleri Enstitüsü), 2018.

[6] Krizhevsky, A., Sutskever, I., Hinton, G. E. "Imagenet classification with deep convolutional neural networks." *Advances in Neural Information Processing Systems*, vol. 25, 2012.
[7] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., et al. "Going deeper with convolutions." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, 2015.

[8] Simonyan, K., Zisserman, A. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556*, 2014.

[9] He, K., Zhang, X., Ren, S., Sun, J. "Deep residual learning for image recognition." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

[10] Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K. Q. "Densely connected convolutional networks." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4700–4708, 2017.

[11] Fei-Fei, L., Fergus, R., Perona, P. "One-shot learning of object categories." *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 28, No. 4, pp. 594–611, 2006.

[12] Paeedeh, N., Pratama, M., Ma'sum, M. A., Mayer, W., Cao, Z., Kowlczyk, R. "Cross-domain few-shot learning via adaptive transformer networks." *Knowledge-Based Systems,* Article ID 111458, 2024.

[13] Zhao, P., Wang, L., Zhao, X., Liu, H., Ji, X. "Few-shot learning based on prototype rectification with a self-attention mechanism." *Expert Systems with Applications*, Article ID 123586, 2024.

[14] Snell, J., Swersky, K., Zemel, R. "Prototypical networks for few-shot learning." *Advances in Neural Information Processing Systems*, pp. 4077–4087, 2017.

[15] Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H., Hospedales, T. M. "Learning to compare: Relation network for few-shot learning." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1199–1208, 2018.

[16] Wang, Y., Chao, W. L., Weinberger, K. Q., Van Der Maaten, L. "Simpleshot: Revisiting nearest-neighbor classification for few-shot learning." *arXiv preprint arXiv:1911.04623*, 2019.

[17] Gülcü, A., Alkan, M. "Az Örnekle Öğrenme Problemleri için MAML ve ProtoNet Algoritmalarının İncelenmesi." *Avrupa Bilim ve Teknoloji Dergisi*, No. 21, pp. 113–121, 2021.

[18] Işık, G. "Improving Plant Disease Recognition Through Gradient-Based Few-shot Learning with Attention Mechanisms." *Journal of the Institute of Science and Technology*, vol. 13, No. 3, pp. 1482–1495, 2023.

[19] Argüeso, D., Picon, A., Irusta, U., Medela, A., San-Emeterio, M. G., Bereciartua, A., Alvarez-Gila, A. "Few-Shot Learning approach for plant disease classification using images taken in the field." *Computers and Electronics in Agriculture*, vol. 175, Article ID 105542, 2020.

[20] Wang, B., Wang, D. "Plant leaves classification: A few-shot learning method based on siamese network." *IEEE Access*, vol. 7, pp. 151754–151763, 2019.

[21] Frikha, A., et al. "Few-shot one-class classification via meta-learning." *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, No. 8, 2021.

[22] Chen, D., Chen, Y., Li, Y., Mao, F., He, Y., Xue, H. "Self-supervised learning for few-shot image classification." *ICASSP 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* pp. 1745–1749, IEEE, June 2021.

[23] Krenzer, A., Heil, S., Fitting, D., Matti, S., Zoller, W. G., Hann, A., Puppe, F. "Automated classification of polyps using deep learning architectures and few-shot learning." *BMC Medical Imaging*, vol. 23, No. 1, Article ID 59, 2023.

[24] Liu, B., Yu, X., Yu, A., Zhang, P., Wan, G., Wang, R. "Deep few-shot learning for hyperspectral image classification." *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, No. 4, pp. 2290–2304, 2018.

[25] Kang, D., Cho, M. "Integrative few-shot learning for classification and segmentation." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9979–9990, 2022.

[26] Hu, Y., Gripon, V., Pateux, S. "Leveraging the feature distribution in transfer-based few-shot

learning." *International Conference on Artificial Neural Networks*, pp. 487–499, Springer, Cham, September 2021.

[27] Kim, J., Kim, T., Kim, S., Yoo, C. D. "Edge-labeling graph neural network for few-shot learning." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11–20, 2019.

[28] Parkhi, O. M., Vedaldi, A., Zisserman, A., Jawahar, C. V. "Cats and dogs." *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3498–3505, https://doi.org/10.1109/CVPR.2012.6248092, 2012.

[29] Bossard, L., Guillaumin, M., Van Gool, L. "Food-101 – Mining discriminative components with random forests." *European Conference on Computer Vision (ECCV)*, pp. 446–461, https://doi.org/10.1007/978-3-319-10599-4_29, 2014.

[30] Gao, T., Han, X., Zhu, H., Liu, Z., Li, P., Sun, M., Zhou, J. "FewRel 2.0: Towards more challenging few-shot relation classification." *arXiv preprint arXiv:1910.07124*, 2019.

[31] Singh, P., Mazumder, P. "Dual class representation learning for few-shot image classification." *Knowledge-Based Systems*, vol. 238, Article ID 107840, 2022.

[32] Ren, H., Cai, Y., Lau, R. Y., Leung, H. F., Li, Q. "Granularity-aware area prototypical network with bimargin loss for few-shot relation classification." *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, No. 5, pp. 4852–4866, 2022.

[33] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L. "ImageNet: A large-scale hierarchical image database." *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, https://doi.org/10.1109/CVPR.2009.5206848, 2009.

[34] Tan, M., Le, Q. "EfficientNet: Rethinking model scaling for convolutional neural networks." *International Conference on Machine Learning*, pp. 6105–6114, PMLR, May 2019.

[35] Howard, A., Sandler, M., Chu, G., Chen, L. C., Chen, B., Tan, M., et al. "Searching for MobileNetV3." *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1314–1324, 2019.