RESEARCH ARTICLE

Zero-Shot Transfer Learning Framework for Plant Leaf Disease Classification

R. SATYA RAJENDRA SINGH[ID], **(Member, IEEE),**
AND RAKESH KUMAR SANODIYA, (Member, IEEE)
Indian Institute of Information Technology Sri City, Sri City, Chittoor 517646, India

Corresponding author: Rakesh Kumar Sanodiya (rakesh.s@iiits.ac.in)

ABSTRACT Agriculture's pivotal role in sustaining livelihoods and driving economic growth is widely recognized, yet various challenges like the adverse effects of climate change and limited resource availability hinder its productivity. Notably, plants are susceptible to various viruses and bacteria, impacting yield and food security. The emergence of deep learning, particularly convolutional neural networks (CNNs), has transformed agriculture by facilitating tasks such as disease detection. However, a significant challenge arises from the often unrealistic assumption that training and testing data share the same distribution. To address this, domain adaptation and transfer learning techniques have been employed, bridging the gap between different data distributions. Therefore, a novel framework named 'Zero-Shot Transfer Learning' is introduced. This addresses the challenge of improving classifier performance when trained on a source domain with different classes and tested on a target domain, exemplified by tomato and potato datasets. More specifically, in this framework, we include different CNN models along with techniques such as data augmentation, synthetic data generation, and robust discriminative losses, enhancing classifier performance in zero-shot scenarios. Extensive experiments on plant leaf disease classification under the zero-shot Transfer Learning assumption demonstrate the superiority of the proposed framework for effective disease classification. Ultimately, this framework holds the potential to promote crop yield optimization and ensure food security.

INDEX TERMS Agriculture, classification, deep learning, zero-shot transfer learning, disease identification.

I. INTRODUCTION

Agriculture, with its multifaceted role in sustaining livelihoods and driving economic growth, is globally recognized as vital [1]. It not only provides employment opportunities and income but also significantly contributes to a country's gross domestic product (GDP) [2]. Particularly in rural areas, where farming forms the backbone of livelihoods, millions rely on agriculture as their primary income source. As the global population is projected to approach 10 billion by 2050, concerns about food shortages, increased hunger, and escalated food demand emerge [3], [4]. Addressing these challenges becomes paramount given the limitations of agricultural resources.

To address challenges related to disease diagnosis and the impracticality of experts' presence in rural areas, traditional machine learning approaches have been proposed [5], [6], [7], [8]. These approaches focuses on categorizing plant diseases using attributes like texture, type, and color of plant leaf images. Prior research has employed machine learning methods such as k-nearest neighbors [9], random forests [10], support vector machines [11], and k-means clustering [12] for supervised or unsupervised classification after preprocessing steps to remove backgrounds and isolate infected regions. Notably, these methods have been criticized for their performance limitations and evaluation primarily in controlled laboratory settings.

Therefore, recent years have seen a surge in deep learning methods, particularly convolutional neural networks (CNNs), addressing various agricultural tasks. These CNN-based approaches have garnered significant attention and success
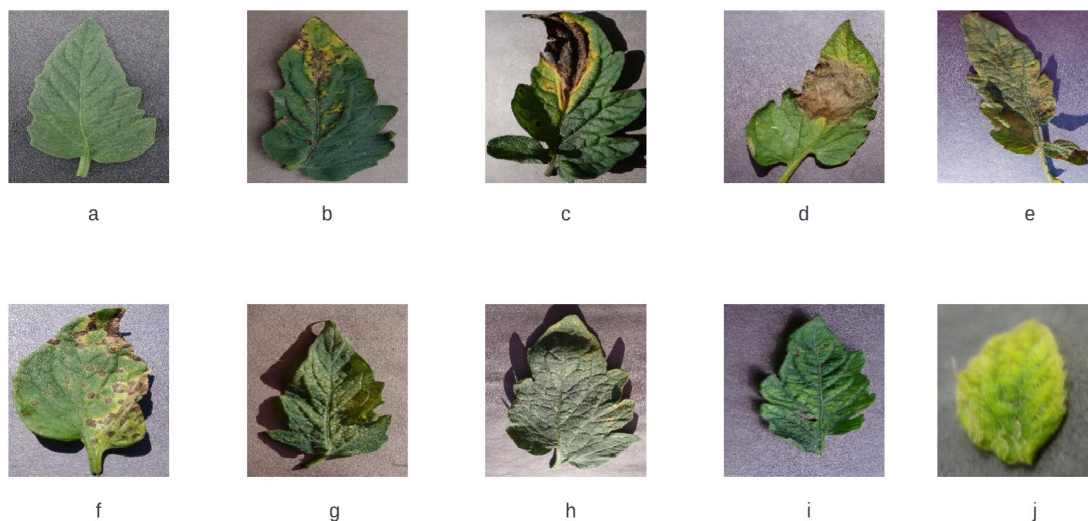
The associate editor coordinating the review of this manuscript and approving it for publication was Donato Impedovo[ID].

FIGURE 1. Tomato plant leaf samples (a) Healthy, (b) Bacterial spot, (c) Early blight, (d) Late blight, (e) Leaf mold, (f) Septoria leaf spot, (g) Two spotted spider mite, (h) Target spot, (i) Mosaic virus, (j) Yellow leaf curl virus.

across diverse agricultural tasks, ranging from pinpointing plant diseases to detecting weeds and classifying crop pests [13], [14]. Notably, CNNs have emerged as a favored architectural choice for various image and video analyses, encompassing pivotal computer vision assignments like image recognition, object detection, and segmentation.

The open-source agriculture datasets have become available, their quality and quantity often do not suitable for real world applications due to lack of robustness in diverse real world environments. To address these difficulties, adopting data augmentation can be helpful. A notable challenge within the domain of deep learning methods revolves around their reliance on annotated data. To effectively tackle this hurdle, researchers have embraced the utilization of data augmentation strategies, encompassing both manual and synthetic methodologies. In the context of manual data augmentation, a rigorous process unfolds, encompassing the creation of images through techniques such as cropping, rotation, and flipping. This augmentation process seamlessly integrates a profusion of supplementary insights into the model, thereby endowing it with the capacity to extract additional attributes that were previously non-existent [15]. Equally significant, the sphere of synthetic data augmentation involves the generation of artificial images, accomplished through well-established generative models like Generative Adversarial Networks (GANs) [16]. This advanced approach facilitates the production of images imbued with diverse styles and contextual settings, thus profoundly enriching the model's capability to comprehend an expanded spectrum of environmental attributes during its training regimen. In the pursuit of enhancing deep learning models even further, a novel approach has been introduced, focusing on the preservation of discriminative information [17]. This strategy entails the minimization of distances between samples from similar classes, while concurrently maximizing the distances

between samples from dissimilar classes. Despite their potential, deep learning approaches assume training and testing data from the same distribution, which may not hold true in real-world scenarios [18], [19]. Domain adaptation and transfer learning emerge as solutions to this distribution shift challenge [18], [20], [21], [22]. Domain adaptation minimizes distribution discrepancies between training and testing data, while transfer learning leverages pre-trained models to enhance performance in new domains. Domain adaptation is used to improve the performance of a model on a target domain by capitalizing on knowledge from a related source domain, even target domain has different data distributions. Domain adaptation is particularly useful in scenarios where collecting labelled data in the target domain is exorbitant or impractical [23], [24].

Nowadays, transfer learning has gained significant attention and has been successfully applied in various domains, including precision agriculture, natural language processing, computer vision, and robotics [25], [26], [27]. Its application has shown promising results in addressing real-world problems across multiple fields such as industrial, security and surveillance, healthcare, agriculture, automobile, and finance. As a result, transfer learning is increasingly recognized as a vital approach for tackling practical challenges. Within the realm of transfer learning, two specialized techniques are zero-shot learning [28], [29], [30], [31] and few-shot learning [32], [33]. These approaches address scenarios where there is limited or no labeled data available in the target domain.

Zero-shot learning is a formidable challenge involving the recognition or categorization of previously unseen objects or classes in a target domain [34], [35], [36]. The model, trained on a labeled source domain, aims to extend its grasp to unfamiliar classes in the target by using extra information like semantic attributes, bridging the gap between the two

domains. This enables the model to predict new classes without direct training. This approach is valuable for expansive or open-world classification tasks. In contrast, few-shot learning addresses knowledge acquisition from a limited set of annotated examples in the source domain [32]. Rather than relying on an extensive dataset, the model learns from scant labeled examples per class. Its objective is to adeptly generalize to new classes or instances in the target domain with minimal supervision. Often using meta-learning, the model swiftly adapts to new tasks by leveraging past knowledge. This empowers the model to generalize effectively and make accurate predictions with minimal labeled examples.

In this paper, we address one of the most critical challenges, which is the identification of diseases within a domain where we lack any labeled examples, making traditional model training impossible. However, we do have access to another domain that contains labeled samples for different classes. Leveraging these labeled samples from the other domain, we aim to identify classes within our target domain. As illustrated in Figure 2, we are provided with a source domain dataset for tomatoes with three labeled classes: Tomato-late-blight, Tomato-Healthy, and Tomato-Early-blight. Our objective is to apply this knowledge to classify the target domain dataset for potatoes, which consists of three classes: Potato-late-blight, Potato-Healthy, and Potato-Early-blight. The task at hand falls under zero-shot learning, one of the most challenging problems in disease identification field.

In this work, we introduce a novel framework called 'Zero-Shot Transfer Learning' to address the problem of improving classifier performance when trained on source domain labeled data with different classes and tested on target domain data. We explore various techniques within this framework, including data augmentation methods such as manual augmentation and synthetic data generation using Generative Models. Additionally, we investigate the benefits of employing robust discriminative losses, such as center loss and triplet losses, to further enhance the classifier's performance.

The contributions of our proposed work can be succinctly summarized as follows:

- We present an innovative framework that harnesses labeled data from the source domain to facilitate accurate classification of diverse classes within the target domain.
- To ensure seamless knowledge transfer between the source and target domains, we explore a range of strategies. These include the integration of pre-trained models, application of data augmentation techniques (both manual and generative), and the incorporation of discriminative methods.
- Through comprehensive experimentation on real-world datasets, we provide empirical evidence showcasing the effectiveness of each strategy integrated within our proposed framework.

II. RELATED WORKS

After conducting a comprehensive literature survey, we have identified five main categories of existing work related to our paper: 1) Traditional machine learning methods, 2) CNN-Based Supervised Learning (CNN-SL), 3) Data Augmentation Methods, 4) Discriminative Methods, 5) Transfer Learning and Domain Adaptation Methods, and 6) Zero-short Learning Methods. These five categories of existing work lay the foundation for our research, and we aim to build upon these insights to propose our innovative framework of Zero-Shot Transfer Learning.

A. TRADITIONAL METHODS

In previous endeavors, a variety of machine learning techniques have been harnessed to address the challenge of identifying plant diseases. Typically, these approaches involve several distinct stages:

- Input Leaf Image: The process commences by considering an input leaf image.
- Pre-processing Techniques: Employing pre-processing techniques to eliminate noise and enhance image quality.
- Segmentation Techniques: Executing segmentation techniques to isolate the region of interest within the leaf image, thereby isolating the affected areas.
- Disease Detection: Employing implemented algorithms to detect diseases present in the segmented regions.
- Feature Extraction Techniques: Extracting features such as Local Binary Patterns (LBP), Histogram of Oriented Gradients (HoG), and Speeded-Up Robust Features (SURF) from the images.
- Feature Selection: Employing techniques like Particle Swarm Optimization (PSO) and Principal Component Analysis (PCA) to select crucial features.
- Classification: Utilizing the selected features to train classifiers such as Support Vector Machines (SVM) and K-Nearest Neighbors (KNN) to perform the final disease classification.

To gauge the effectiveness of disease identification, common performance measures such as Accuracy, Area Under the Curve (AUC), and F-score are often considered.

For instance, Dhingra et al. [37] conducted an in-depth exploration of digital image processing techniques for leaf disease detection and classification. This study comprehensively discusses the performance of disease detection and classification, analyzing state-of-the-art techniques introduced between 1997 and 2016. Singh and Misra [38] proposed an algorithm for image segmentation, which facilitated automatic detection and classification of plant leaf conditions. Chilingaryan et al. [39] delved into research developments over the past 15 years regarding machine learning techniques for accurate crop yield prediction and estimation of nitrogen status. Pydipati et al. [40] introduced the color co-occurrence method (CCM) that combines with statistical classification algorithms to classify diseased and normal cit-
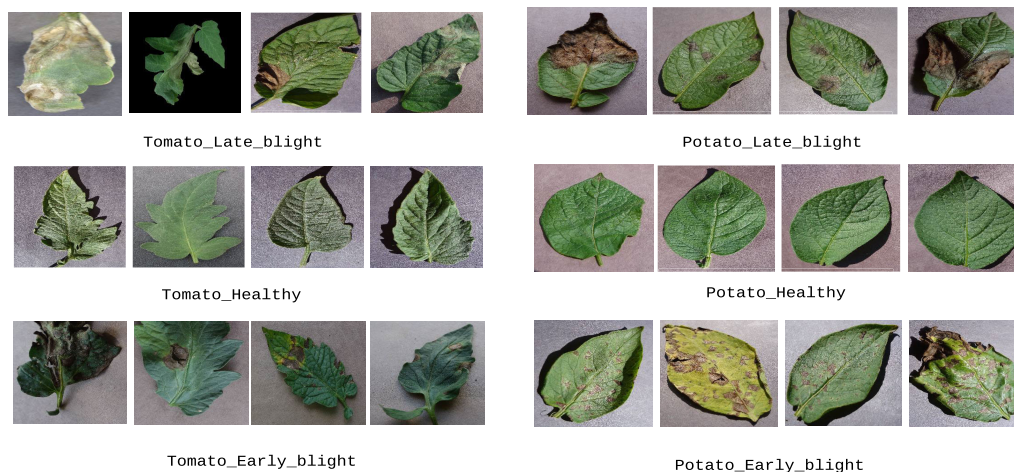
FIGURE 2. Zero short transfer learning: source domain classes, target domain classes.

rus leaves based on texture-based hue, saturation, and intensity (HSI) color features. Camargo and Smith [41] outlined an image processing approach for identifying visual symptoms of plant diseases through exploration of colored images.

He et al. [42] proposed a method for identifying cotton leaf diseases, employing three color models for feature extraction. Tucker and Chakraborty [43] considered thresholding and clustering parameters techniques for identifying and detecting diseases in oat and sunflower leaves. Aduwo et al. [44] devised an automated vision-based analysis system to detect cassava mosaic disease, using color and shape features and various classifiers, including the Naive Bayes classifier. Muthukannam and Latha [45] emphasized the significance of Particle Swarm Optimization (PSO)-based image segmentation for plant leaf disease identification. Jian and Wei [46] introduced a method for identifying diseases in cucumber leaves using Support Vector Machines (SVM) with polynomial, radial basis function, and sigmoid kernel functions. Chandra and Bedi [47] described various approaches of SVM, parameter selection and multiple approaches impact the efficiency of image classification approaches. Tanveer et al. [48] proposed the concept of using Twin SVM (TWSVM) by which used dual hyperplanes which are not parallel to classify the data points. Further Wahab et al. [49] implemented the approach in which image segmentation is vital, instead of using entire image, parrt of image is considered for detecting diseases in chilli plants. Kour and Arora [50] proposed pre-requisite step before applying SVM in Particle swarm optimization based SVM of the plants segmentation and classification.

B. CNN-BASED SUPERVISED LEARNING (CNN-SL)
In recent years, the cutting-edge approach of using convolutional neural networks for supervised learning (CNN-SL) has found widespread application across various agricultural domains. The CNN models varies depending upon the number of layers and number parameters used in millions as

shown in Table 1 These applications encompass tasks such as identification, classification, detection, quantification, and prediction. In a notable study by Mohanty et al. [51], the efficacy of this technique was demonstrated through the training of two renowned deep convolutional neural networks: AlexNet and GoogLeNet. Their goal was to discern 14 distinct crop species and identify 26 diseases within the PlantVillage dataset. Impressively, GoogLeNet emerged as the standout performer, achieving an exceptional accuracy of 99.35%. However, the robustness of these models was subsequently tested against two separate validation datasets, leading to a significant drop in overall accuracy to a mere 31%. This outcome highlights a notable disparity between the models' performance on the initial dataset and their ability to generalize to other datasets.

Utilizing a VGG-19 based Convolutional Neural Network, the study attains a 95.6% accuracy in automated plant disease classification, leveraging data from PlantVillage. The model, deployed on an Android app, showcases real-time leaf disease detection across 13 plant species, underscoring its robust potential for accurate agricultural disease management. A pioneering hybrid model, combining Convolutional Autoencoder (CAE) and CNN, achieving remarkable results in plant disease detection. Employing the model on Bacterial Spot disease in peach plants, it attains 99.35% training and 98.38% testing accuracy, utilizing only 9,914 training parameters, offering improved precision and efficiency in automatic disease identification. Thakur et al. [52] elevated the importance of machine learning approaches for vision based approaches for plant disease identification. The authors pointed out the datasets availability and influence in models performance. Hassan and Maji [53] study firmly ratified the need of technology to replace manual actions in decision making, from inception of image capturing to model decision making for image classification by CNN models.

Combining UAVs with deep neural networks (DNNs), the study employs Inception_v1, Inception_v2, Inception_v3, and LeNet models, with Inception_v2 achieving 100%

TABLE 1. State of art CNN models.

Model	Proposed Year	No of layers	No of Parameters
AlexNet	2012	8	62.3 M
DenseNet121	2016	121	6.99 M
EfficientNet_b0	2019	66	5.3 M
GoogLeNet	2014	22	5.64 M
MnasNet1_0	2018	39	4.4 M
MobileNet_v1	2017	28	3.25 M
Resnet 50	2015	50	23.59 M
ShuffleNet_v1	2018	27	5.4 M
VGG-16	2014	16	134.42 M
VGG-19	2014	19	143.67 M

accuracy. Yield assessment, integrating classification outcomes and expert estimations, yields 97.93% precision for Inception_v2.

C. DATA AUGMENTATION METHODS

Generative Adversarial Networks (GANs) are a powerful class of generative models that operate in a unique adversarial framework. GANs consist of two main components: a generator and a discriminator. The generator's primary role is to create realistic data, such as images, videos, or audio, from random noise or latent representations. On the other hand, the discriminator acts as a binary classifier, attempting to distinguish between real data from the dataset and data generated by the generator. GANs have demonstrated impressive capabilities in various applications, such as generating high-quality images, creating realistic video sequences, and generating audio signals that sound authentic. They have also been used for data augmentation, style transfer, and image-to-image translation tasks. GANs have had a significant impact on the field of generative modeling, leading to a new era of creative and realistic data synthesis [54]. The model executed on concatenated dataset generated with data augmentation over existing data, will enhance the performance of the system by minimizing over fitting issues [55]. cGANs generate artificial images of maize and common weeds (Charlock, Fat Hen, Shepherd's Purse, and Small-flowered Cranesbill) to expand the dataset [56]. Zeng et al. implemented methods with support of GANs to detect severity in infected leafs in citrus plants [57]. The Data augmentation will influence the model efficiency even in the circumstances having small datasets of annotated images used for segmentation in plant disease. The data generated by augmented methods will support to minimize the efforts to create pixel wise segmented annotations which can be implemented in real time applications [58]. The fidelity of synthetic images is evaluated using t-SNE visualization, and improved performance is demonstrated in crop/weed classification models using transfer learning (CNN) and feature extraction (SVM, LDA) techniques.

D. DISCRIMINATIVE METHODS

In the realm of literature, various discriminative loss functions such as hinge loss, triplet loss, center loss, and more, have been explored to enhance the discriminative capabilities

of classifiers. For example, Center loss is used in conjunction with softmax cross-entropy loss for face recognition tasks. It encourages the features of each class to cluster around their class-specific center, making the intra-class variations smaller and enhancing discriminability [59]. Triplet loss is considered in triplet networks for tasks like image retrieval. It encourages embeddings from the same class to be closer to each other and farther from embeddings of other classes, thus improving the discriminative power of the model [60]. The hinge loss is commonly used in Support Vector Machines (SVM) for binary classification. It penalizes misclassifications and pushes the model to ensure that the correct class score is higher than the incorrect class score by a certain margin [61]. Margin ranking loss is employed in siamese networks or triplet networks used for tasks like face recognition or similarity learning [62]. The loss compares the similarity scores between anchor-positive pairs and anchor-negative pairs, aiming to maximize the margin between them. Similar to margin ranking loss, contrastive loss is used in siamese networks to learn embeddings that project similar samples close together and dissimilar samples farther apart [63].

Zhang and Zhang [64] introduced Orthogonal Locally Discriminant Projection for classification of Plant Leaf Diseases. Argueso et al. [65] introduced a few-shot learning methodology for the classification of plant diseases through field images. Their study employed three distinct CNN architectures to construct two foundational models, including a Triplet network and a sophisticated Deep Adversarial Metric Learning (DAML) technique. The outcomes of their assessment revealed that a foundational model trained on an extensive collection of source field images could be fine-tuned to effectively classify novel diseases using only a limited quantity of images. Fan et al. [17] presented a novel approach involving deep feature descriptors combined with traditional handcrafted features through feature fusion for enhanced plant leaf image analysis. The integration includes center loss to improve the distinctiveness of fused features, ensuring compactness within classes and separation between them. Experimental validation on three datasets demonstrates significant classification accuracies: 99.79%, 92.59%, and 97.12%.

E. TRANSFER LEARNING AND DOMAIN ADAPTATION

Yin et al. introduced a transfer learning-based model for detecting diseases and pests in Hot peppers [66]. In their study, they delved into eight pre-trained models, including VGG16, VGG19, and ResNet50, to extract intricate features from images. Rangarajan et al. utilized pre-trained VGG16 and AlexNet models to successfully identify six distinct tomato diseases and pests, attaining impressive recognition accuracies of 97.49% for both models [67]. Nsumba et al. employed the pre-trained MobileNet model and a limited dataset to achieve a noteworthy 96% accuracy in detecting cow-pea diseases [68]. Ramcharan et al. employed advanced

TABLE 2. Comparative study used on dataset for different architectures with discriminator and data augmentation.

Author and Year	Dataset Used	Model	Testing on Unknown Classes	Discriminator		Data Augmentation	
				Centre Loss	Triplet Loss	Manual	Synthetic
Rehana et al. [97]-2023	Plant Village	VGG 16	NO	NO	NO	NO	NO
Jha et al. [111]-2023	Plant Village	VGG-16 InceptionV3	NO	YES	NO	YES	NO
Prasad et al. [110]-2022	Plant Pathology dataset	EfficientNet-B0	YES	NO	NO	NO	YES
Olaniyi et al. [104]-2022	Multi-Class/Modal Datasets	YOLO	YES	NO	NO	YES	YES
Paymode et al. [98]-2022	Plant Village	VGG 16	YES	NO	NO	YES	YES
Schuler et al. [106]-2022	Plant Village	Inception V3	Yes	NO	NO	NO	NO
Tiwari et al. [100]-2021	Plant Village iBean leaf image dataset Citrus leaf images Rice leaf images	DenseNet	YES	NO	NO	NO	NO
Katafuchi et al. [103]-2021	DR2, PlantVillage, MVTec, Cloud datasets.	ResNet MobileNet-v2	NO	NO	NO	NO	NO
Bollis et al. [109]-2020	Citrus Pest Benchmark	ResNet50 EfficientNet-B0, MobileNet-v2	YES	NO	NO	NO	NO
Esgario et al. [105]-2020	Arabica Coffee Leaves	ResNet 50	No	NO	NO	Yes	No
Nazki et al. [108]-2020	Plant Village	ResNet50	NO	NO	NO	NO	YES
Rubanga et al. [102]-2020	In-house data collection	VGG16, VGG19, ResNet, Inception-V3	NO	NO	NO	Yes	NO
Brahimi et al. [107]-2019	Plant Village	VGG16, Auto Encoder	NO	NO	NO	NO	NO
Kaur et al. [99]-2019	Plant Village	ResNet 101	NO	NO	NO	YES	YES
Gandhi et al. [101]-2018	Plant Village	Inception V3 MobileNet	NO	NO	NO	NO	YES
Proposed Method	Plant Village	State of art models	**YES**	**YES**	**YES**	**YES**	**YES**

transfer learning techniques to detect cassava diseases, achieving an accuracy rate of 95% [69]. Wang et al., utilized transfer learning, employing VGG16, VGG19, Inception V3, and ResNet50 models to discern different stages of leaf health, achieving a commendable 90.4% accuracy primarily with the VGG16 model [70]. Llorca et al., in their research, employed the Inception V3 model to recognize pets and tomato diseases, securing a notable accuracy of 88.8% [71]. Vallabhajosyula et al. proposed the state of art performance technique which implemented by using weighted ensemble neural networks for automatic plant disease approaches [72]. Zhao et al. proposed the method in domain adaptation, with vast unlabelled data to limited label data fine turning the model and got better performance and accuracy [73]. Ma et al. by implementing unsupervised domain adaptation approach for better performance, proposed the method for crop yield prediction in which features extracted from multiple domain sources [74]. Magistri et al. proposal implementation majorly unitized segmentation in domain adapation for real world crop monitoring application model. Instead of treating the entire field as a single unit, divided into a semantic way to reach out effectively to get better results [75]. In Table 2 summary, provided the information regarding authors worked on the Plant Village dataset, employed variations in model architectures, testing approaches, and the utilization of techniques such as data augmentation and loss functions. The table summarised the approach followed by authors regarding encompassing manual data augmentation and/or synthetic augmentation, employed discriminators like Centre Loss and Triplet Loss, model tested on unknown data or not.

F. ZERO-SHOT LEARNING METHODS

In the realm of literature, the majority of efforts dedicated to zero-shot learning have been centered around tasks such as object recognition and segmentation within real-world datasets. Surprisingly, none of these endeavors have delved into the intricate challenge of zero-shot learning applied to disease classification. For example, in Zero-shot Learning (ZSL), semantic embedding spaces are utilized as a means of transferring knowledge. Most existing ZSL [76], [77], [78] employ attribute spaces as their semantic embedding spaces. However, using attribute spaces to represent object classes necessitates the manual definition of an attribute ontology, which can limit the effectiveness of attribute space-based ZSL methods. To address this limitation, [79], [80], [81] explored the use of semantic word vector spaces as an alternative to attribute spaces. There are some ZSL approaches, which are based on visual-semantic similarity matching [82]. The issue of domain shift in Zero-shot learning (ZSL) was initially reported in [35] and referred to as the projection domain shift problem. This problem was addressed by introducing transduction multi-view embedding. A new zero-shot learning (ZSL) technique for unsupervised domain adaptation was presented by Hou et al. [83]. To tackle the problem of projection domain shift, the authors introduced a novel regularized sparse coding framework. Reference [84] presented the zero-shot learning for domain adaptation, where the test instances are restricted to be only from unseen classes. The article cited as [85] seeks to tackle the issue of generalized zero-shot learning, as outlined in [84]. This problem arises in a more practical scenario where

test instances may fall into any category, whether it is a known or unknown class. Li et al. [80] proposed a method for generalized zero-shot domain adaptation called TUPL (Target Unseen class Prototype Learning). In this approach, samples from both domains are projected into a shared subspace, ensuring that samples from the same class are close together to address domain differences. Attribute-Based Zero-Shot Learning, the model is trained on a source task with labeled data and learns to associate attributes or semantic descriptions with the classes. During zero-shot transfer, the model uses these attribute vectors to predict the target classes, even without any labeled data from the target task [86].

Embedding-based methods leverage shared embeddings to transfer knowledge across tasks [87]. By learning a common embedding space during pre-training, the model can map new inputs to the same space and perform well on unseen tasks. Model-based methods zero-shot learning, a generative model is trained during pre-training to represent the data distribution [88]. During testing, the generative model can be used to synthesize samples for unseen classes. Generative adversarial networks (GANs) can be used in zero-shot learning to generate samples for unseen classes. The generator is trained on seen classes during pre-training, and then it can generate samples for unseen classes during testing [89]. Tsai et al. [90] proposed Deep Domain Adaptation approach, which uses privileged information from task-irrelevant dual-domain pairs. Zhang et al. [91] proposed a novel method for zero-shot domain adaptation (ZSDA) that involves training domain-invariant semantic features and task-invariant domain features simultaneously using adversarial learning. The approach aims to learn domain shift domain features in a task-agnostic manner. Meanwhile, [92] investigated the zero-shot scenario in the day-night domain by leveraging prior knowledge obtained from a physics-based reflection model.

TABLE 2 depiction the literature in which various authors worked on Plant village Dataset and considering various objectives like testing the model on unknown classes, usage of discriminators like Center Loss, Triplet Loss and Data augmentation approaches either manual or synthetic data generators compared with our proposed method. TABLE 2, it can be seen that none of the existing approaches except our proposed approach satisfy all the important objectives required for improving the performance of the model on the unknown class samples.

III. METHODOLOGY

Summary of the Zero-Shot Transfer Learning Framework Pipeline is shown in Figure 3. This diagram illustrates a comprehensive architecture designed for the classification of plant leaf diseases. The process encompasses several key steps, including data preparation, augmentation, normalization, integration of a pre-trained model, consideration of loss functions, model training, and final model testing. The pipeline is structured into seven distinct phases: Dataset Preparation (Step 1): This phase involves the selection and

TABLE 3. Plant village dataset images category wise and crop type.

S. No	category	Type	Total images	Dieses	Healthy
1.		Apple	3172	1527	1645
2.		Blueberry	1502	-	1502
3.		Cherry	1906	1052	854
4.	Fruits	Grape	4063	3640	423
5.		Orange	5507	5507	-
6.		Peach	2657	2291	360
7.		Raspberry	371	-	371
8.		Strawberry	1565	1109	456
9.		Corn	3852	2690	1162
10.		Bell Pepper	2475	997	1478
11.	Vegetables	Potato	2152	2000	152
12.		Soybean	5090	-	5090
13.		Squash	1835	1835	-
14.		Tomato	18162	16570	1592

preparation of datasets for the zero-shot learning process. Augmentation Techniques (Step 2): Step 2 focuses on the application of various augmentation techniques to enhance the diversity and richness of the dataset. Normalization Strategies (Step 3): Step 3 highlights the implementation of normalization techniques to ensure consistent and standardized input data. Pre-trained CNN Models (Step 4): In Step 4, a range of pre-trained Convolutional Neural Network (CNN) models are presented for feature extraction. Loss Function Integration (Step 5): Step 5 involves the integration and comparison of different loss functions, contributing to the overall effectiveness of the model. Fine-tuning the Model (Step 6): Step 6 details the fine-tuning process, where the model is adjusted to improve its performance and adapt it to the specific task. Ultimately, the pipeline concludes with the Evaluation of Fine-tuned Models (Step 7) on the target data, assessing the model's ability to accurately classify plant leaf diseases.

A. STEP-1: DATA PREPARATION

Plant Village Dataset: Plant Village dataset consists of 38 classes, classes classified based on type of crop, type of disease. The crop type under fruits category Apple, Blueberry, Cherry, Grape, Orange, Peach, Raspberry, Strawberry. In vegetables category Corn, Bell Pepper, Potato, Soybean, Squash, Tomato. The classes classification based on type of diseases like fungal, bacterial, oomycete, viral and mite. Apart from above mentioned the dataset consists of classes of healthy leaves. The Plant Village dataset consists of 54,303 images from 14 different crop types including 26 disease categories and 14 healthy leafs classes shown in Table 3.

To evaluate the performance of Zero short Transfer learning model, experiments with different settings were performed. The experiments performed on images of group of classes treated as source domain and tested on images of group of classes treated as target domain. Initially source data with different discriminators executed. Later, apart from original images, data augmented model generated instances added and treated as source data one set of training will be performed. Later data augmented images added to original instances and performed another set of training performed.
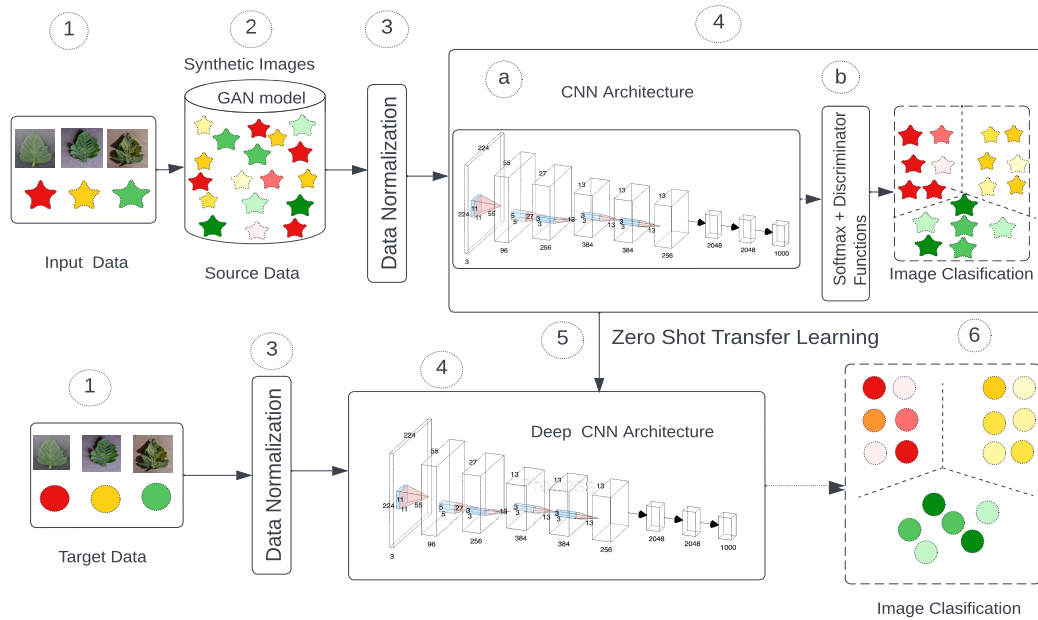
FIGURE 3. Proposed architecture.

TABLE 4. Task oriented class wise image distribution.

Task	Source Domain		Target Domain	
	Class Name	# of Instances	Class Name	# of Instances
T-1	TLB	1909	PLB	1000
	TH	1000	PH	152
	TEB	1590	PEB	1000
T-2	PLB	1000	TLB	1909
	PH	152	TH	1000
	PEB	1000	TEB	1590

The classes from Plant Village dataset used for zero short transfer learning experiments are Tomato Light Blight(TLB), Tomato Healthy(TH), Tomato Early Blight(TEB), Potato Learly Blight(TEB), Potato Healthy(PH) and Potato Early Blight(PEB). Each class having different image instances as shown in Table 4. The experimental setup divided into task wise, Task-1 consists TLB,TH, TEB classses images as source, PLB, PH, PEB classes images as target. In Task-2 PLB, PH, PEB classes images as source and TLB, TH, TEB classes instances as target. To distinguish each class instances are represented with unique color as shown in Figure 3 Step-1.

B. STEP-2: AUGMENTATION TECHNIQUES
In the realm of data augmentation, we delve into two primary strategies: 1) Manual Image Generation, and 2) Synthetic Image Generation.

1) MANUAL IMAGE GENERATION
we focused to generate new images from original images. The original images from different classes be perform data augmentation. In this process, initially we take original image rotated with specific angle and saved the generated image. The new generated image corners may be out of

dimension then edges cropped to fit the fixed dimension. Within this technique, we explore elementary operations like rotation, cropping, and flipping. These subtle alterations introduce variations to images, enhancing the model's ability to generalize by learning from diverse viewpoints of the same object or scene. In the proposed approach, we have taken original image and applied fixed angles starting with 5, multiples of 5 like 10, 15, 20 till 90, the new images generated will be cropped to fixed dimension as of original image as shown in Figure 4.

2) SYNTHETIC IMAGE GENERATION
This process involves crafting entirely new images based on the inherent patterns and characteristics of the original dataset. This approach often employs more sophisticated methodologies, such as generative adversarial networks (GANs) or image-to-image translation models. These techniques yield exceptionally realistic images closely resembling the original data as shown in Figure 5.

In scenarios where the source domain comprises numerous classes but only a handful of samples per class, and this data is considered for domain adaptation, the learned model might struggle to generalize effectively for the target domain. To tackle this challenge, Generative Adversarial Networks (GANs) [93] have emerged as a solution. GANs are a subset of modern neural networks utilizing two opposing models: the Generator ($\mathcal{G}$) and the Discriminator ($\mathcal{D}$) during training. The Generator ($\mathcal{G}$) endeavors to deceive the Discriminator ($\mathcal{D}$) by producing output images akin to the provided input dataset. Meanwhile, the Discriminator strives to enhance its discernment by classifying the generated images as authentic or counterfeit.

In this work, we explore variant of GAN like Conditional Generative Adversarial Networks (cGANs) [94]. This
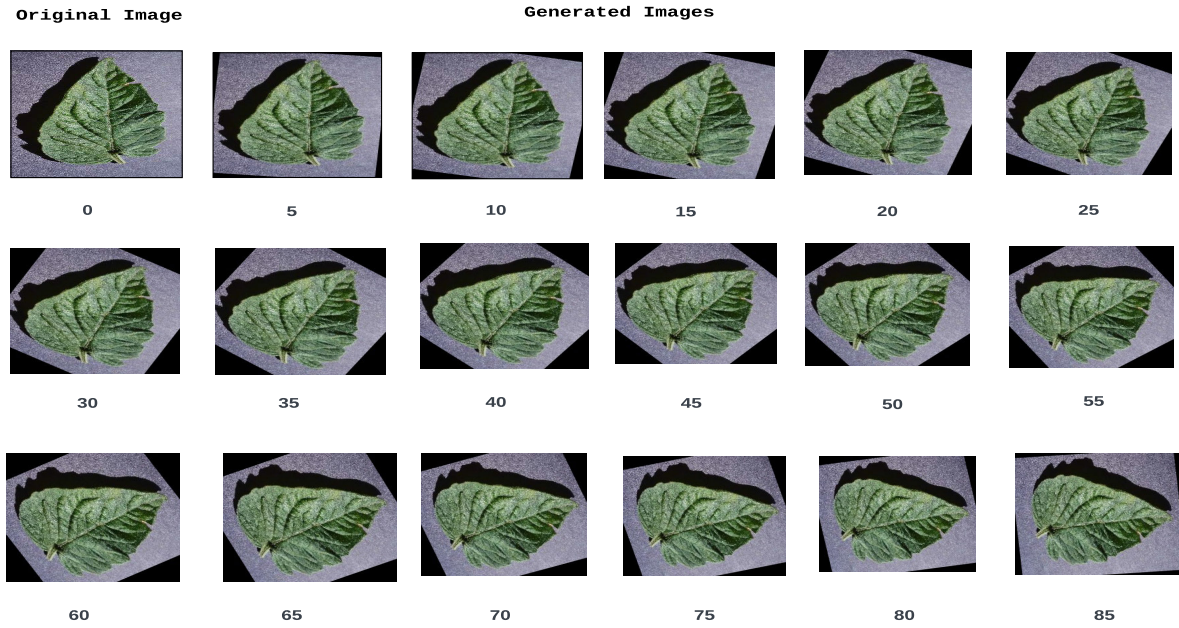
FIGURE 4. Data augmented images generated by different angle rotation angle indicated at bottom of each image and cropping the original image of Tomato Healthy leaf class (first row, first column).
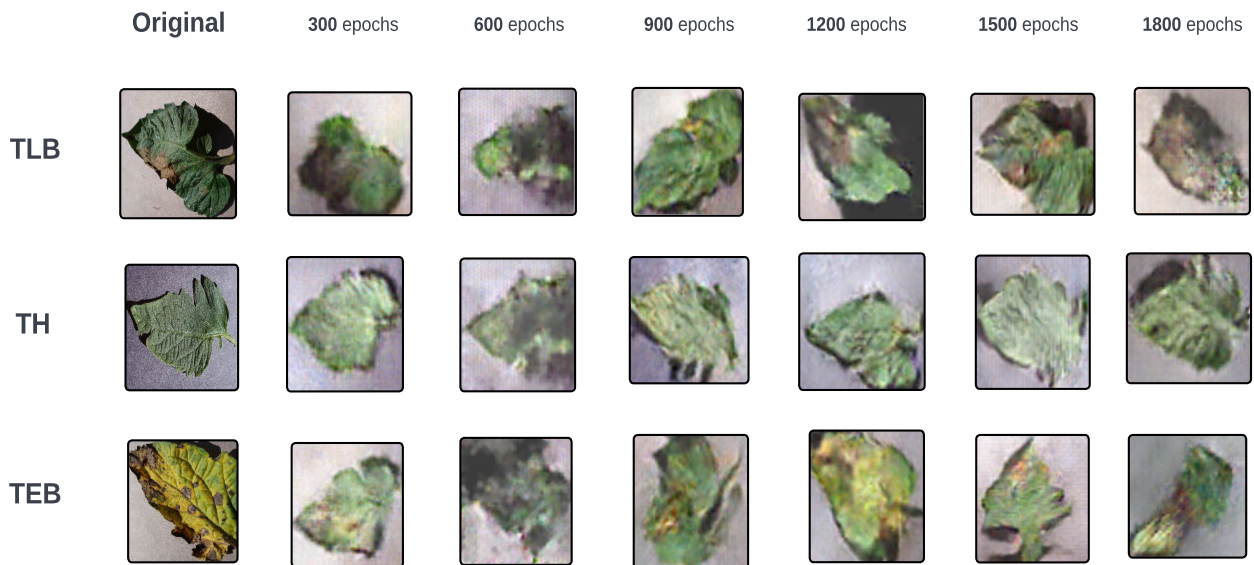

FIGURE 5. Data augmented images generated by generative adversarial networks the original image of Tomato Light Blight class, Tomato Healthy and Tomato Early Blight class with generated images for different no of epochs.

innovative model aims to generate more lifelike synthetic images for the source domain, bolstering the model's generalizability to the target domain. The loss function for the cGANs model is articulated in Eq. (1):

$$\min_{\mathcal{G}} \max_{\mathcal{D}} V(D, G) = \mathbb{E}_{x \sim p_{data}(x)}[log \, D(x|y)]$$
$$+ \mathbb{E}_{z \sim p_z(z)}[log(1 - D(G(z|y)))] \quad (1)$$

This equation succinctly captures the interplay between the generator and discriminator, emphasizing the importance of their balance.

Through cGANs, we empower ourselves to generate synthetic data that is conditioned on labeled data from the source domain. This is achieved by the dynamic interplay between the authentic source data and the synthetically generated data.

C. STEP-3: NORMALIZATION STRATEGIES
Given that real-world data is sourced from diverse resources, each with its individual constraints, it's essential to acknowledge the potential for noise within the dataset. To address this, we employ normalization techniques. These techniques

play a crucial role in standardizing the data and bringing it to a uniform distribution, thereby facilitating more accurate and insightful analyses. Consequently, we are considering the utilization of min-max normalization techniques before supplying the data to the CNN Model.

The normalization process can be represented by the equation:

$$x_{norm} = \frac{x}{\max(X)} \qquad (2)$$

In this equation, x represents a data sample, and X corresponds to a feature vector. This normalization approach aims to scale the data appropriately, ensuring that each feature contributes effectively to the model's training process.

D. STEP-4: PRE-TRAINED CNN MODELS

Consider a real-world scenario in which you might encounter a task with very limited or unavailable labeled data. In such situations, developing a deep learning model becomes impractical or even impossible. To address this challenge, transfer learning approaches have been introduced. Transfer learning leverages the concept of pre-trained models, specifically Convolutional Neural Network (CNN) architectures trained on abundant labeled data like ImageNet. The effectiveness of this approach largely hinges on the choice of an appropriate CNN model. This selection depends on the specific requirements of the problem at hand. For instance, if the model's evaluation is constrained by low computational resources, opting for a CNN model with fewer parameters, such as MobileNetV2, proves advantageous. Conversely, if resource availability allows, selecting a higher parameter model like VGG19 could be beneficial. In this paper, we delve into both types of models for our experiments, facilitating a comprehensive comparative analysis.

E. STEP 5: INCORPORATING THE LOSS FUNCTION

In the training of the CNN model, we incorporate the cross-entropy loss function as follows:

$$\mathcal{L}_s = -\sum_i y_i \log(\hat{y}_i), \qquad (3)$$

where y_i represents the true label and $\hat{y}_i$ signifies the predicted label for the ith sample.

To enhance the classification capability of the model, we introduce two additional discriminative components to the classifier.

1) TRIPLET LOSS

The motivation behind adopting the triplet loss is twofold: it encourages the projection of all images of a single subject onto a single point in the embedding space, while simultaneously enforcing a margin between pairs of images of the same subject and all other subjects.

In the context of the triplet loss, for a specific subject, the distance between an image x_i^a (anchor) and all other images

x_i^p (positive) of the same subject is smaller than the distance to any image x_i^n (negative) of any other subject.

The triplet loss function is defined as follows:

$$\left\| f(x_j^a) - f(x_j^p) \right\|_2^2 + \Delta < \left\| f(x_j^a) - f(x_j^n) \right\|_2^2,$$
$$\forall (f(x_j^a), f(x_j^p), f(x_j^n)) \in \mathcal{T}, \qquad (4)$$

where $\mathcal{T}$ is the set of all possible triplets in the training data with a cardinality of M, and α denotes the margin between positive and negative pairs. The minimized loss is formulated as:

$$\mathcal{T}_L = \sum_{i=1}^{M} [\left\| f(x_s^a) - f(x_s^p) \right\|_2^2 - \left\| f(x_s^a) - f(x_s^n) \right\|_2^2 + \Delta]+, \qquad (5)$$

After incorporating the triplet loss with the cross-entropy loss, the final loss function becomes:

$$\mathcal{L} = -\sum_i y_i \log(\hat{y}_i) + \mathcal{T}_L. \qquad (6)$$

2) CENTER-BASED LOSS

The center-based loss was initially introduced for face recognition. We adapt this loss by combining it with the cross-entropy loss for increased discrimination [96].

The process begins by passing the training images through a network pretrained on Imagenet, yielding feature descriptors $f_{y_i}(x_i)$. The center c_k of the kth class is then computed as:

$$c_k = \frac{1}{N_k} \sum_{y_i=k} f_{y_i}(x_i), \qquad (7)$$

Furthermore, the distance d_{ik} between the feature descriptor of each image and each class center c_k is calculated as follows:

$$d_{ik} = \left\| f_{y_i}(x_i) - c_k \right\|_2^2. \qquad (8)$$

The overall center-based loss can be computed as

$$C_L = \sum_{k=1}^{C} d_{ik} \qquad (9)$$

where C is the number of clases. Upon incorporating the center-based loss with the cross-entropy loss, the final loss function becomes:

$$\mathcal{L} = -\sum_i y_i \log(\hat{y}_i) + C_L. \qquad (10)$$

F. STEP 6: TRAINING THE MODEL

We can seamlessly implement the proposed method using mini-batch Stochastic Gradient Descent (SGD) [95]. The total loss is defined as $\mathcal{L} = \mathcal{L}_s + \alpha \mathcal{L}_d$. Here, the cross-entropy is a conventional softmax classifier. $\mathcal{L}_d$ could be either $\mathcal{T}_L$ or $\mathcal{C}_e L$. Both the functions are differentiable with respect to

TABLE 5. Zero short transfer learning with data augmentation using GANs models.

Tasks	Model	Without Augmentation		ZsTL-DAG		
		Without-training	With-training	200 instances	300 instances	500 instances
T-1	VGG16	50.27	71.46	**73.14**	67.93	70.58
	VGG19	49.95	65.42	**68.82**	64.26	67.19
	ResNet18	42.70	52.23	56.64	**56.83**	55.57
	AlexNet	49.21	56.22	**61.80**	57.01	57.85
	GoogLeNet	31.22	54.41	52.18	**54.60**	56.59
	DenseNet121	42.33	49.30	50.32	**55.66**	52.41
	EfficientNet_b0	36.33	66.68	65.38	66.40	**71.46**
	MnasNet1_0	9.61	18.16	**20.95**	18.30	16.72
	MobileNet_V2	45.07	61.43	57.06	59.01	**62.96**
	MobileNet_V3_Small	17.89	46.28	**47.44**	44.28	46.14
	ShuffleNet_v2_x0_5	23.97	49.21	50.74	**51.44**	48.32
T-2	VGG16	40.63	45.20	**50.06**	45.55	46.34
	VGG19	44.92	48.29	**59.73**	52.95	53.04
	ResNet18	38.83	41.87	**44.36**	42.20	43.94
	AlexNet	42.32	**45.03**	42.69	42.47	42.05
	GoogLeNet	39.67	42.14	42.58	42.18	**42.78**
	DenseNet121	42.20	**43.36**	42.74	43.25	42.76
	EfficientNet_b0	40.18	40.18	42.69	40.65	**45.58**
	MnasNet1_0	39.31	**59.94**	57.56	56.59	52.43
	MobileNet_v2	41.07	41.36	42.34	**45.14**	44.25
	MobileNet_v3_Small	35.23	45.05	**48.94**	43.52	42.23
	ShuffleNet_v2_x0_5	27.87	41.94	**42.65**	41.16	41.83

the inputs, allowing the parameters Θ to be updated through standard backpropagation:

$$\Theta_{t+1} = \Theta_t - \eta \frac{\partial(\mathcal{L}_s + \alpha\mathcal{L}_d)}{\partial(x_i)}, \qquad (11)$$

where η represents the learning rate.

G. STEP 7: ASSESSING FINE-TUNED MODELS

After training the model using the training dataset, it's essential to evaluate its performance using the testing dataset. The evaluation is typically done in terms of accuracy, calculated using the following formula:

Accuracy (ACC)

$$= \frac{\text{Number of Correct Predictions}}{\text{Total Number of Cases}} \times 100 \qquad (12)$$

This metric provides insight into how well the model's predictions align with the actual outcomes in the testing phase.

IV. EXPERIMENTATION AND RESULTS

This section presents comprehensive information regarding the experimental setup, data-set statistics, evaluation metrics employed in this study, as well as a series of experiments conducted and their corresponding results. Additionally, a thorough comparative analysis with alternative methods is provided to offer further insights.

A. TRAINING CONFIGURATION

The training setting for experiments, Firstly for data augmentation, the images are used in the same sizes, i.e., 32×32. The mini batch size was 100. We use softmax classifier and Stochastic Gradient Descent optimization with momentum 0.9 and learning rate 0.001.

B. COMPARISON WITH STATE-OF-THE-ART CNN MODELS

In the realm of cutting-edge convolutional neural networks (CNNs), a comparative analysis with state-of-the-art models VGG16 [97], [98], [102], [107], [111], VGG19 [102], ResNet [99], [102], [103], [105], [108], [109], AlexNet, GoogLeNet, DenseNet [100], Inception V3 [101], [102], [106], [111], EfficientNet-B0 [109], [110], MnasNet1_0, MobileNet [101], [103], [109] and ShuffleNet_V2 for their depth and sophisticated architecture, a comprehensive exploration of their performance when confronted with similar datasets as shown in TABLE 2. These models are examined within the context of a shared dataset, incorporation of discriminative abilities and the impact of data augmentation on their feature extraction capabilities.

C. EXPERIMENT SETTINGS

1) ZERO-SHOT TRANSFER LEARNING (ZsTL)

In this setting, we explore pre-trained CNN models in two ways, one without fine-tuning and the other with fine-tuning of existing pre-trained CNN models. In the setting without fine-tuning, we assess the performance of the pre-trained model on the target domain samples directly, without any further adjustments. However, in the fine-tuning stage, we take the existing pre-trained model and fine-tune its weights using source data samples. This fine-tuning process extends for 100 epochs, after which we evaluate the model's performance on the target domain samples.

2) ZsTL WITH DATA AUGMENTATION USING GANs (ZsTL-DAG)

In this setting, we focus on conditional generative adversarial neural networks (CGANs) generated images and existent images as source domain images. The newly generated

TABLE 6. Zero short transfer learning with data augmentation using cropping.

Tasks	Model	Without Augmentation		ZsTL-DAC		
		Without-training	With-training	200 instances	300 instances	500 instances
T-1	VGG16	50.27	71.46	66.91	73.08	**75.38**
	VGG19	49.95	65.42	66.75	65.80	**69.59**
	ResNet18	42.70	52.23	52.59	**56.85**	53.00
	AlexNet	49.21	56.22	53.84	**58.47**	54.77
	GoogLeNet	31.22	54.41	49.13	54.93	**55.97**
	DenseNet121	42.33	49.30	52.89	50.36	**53.30**
	EfficientNet_b0	36.33	66.68	63.82	64.42	**71.53**
	MnasNet1_0	09.61	18.16	23.63	**30.81**	26.27
	MobileNet_V2	45.07	61.43	58.38	**63.05**	57.10
	MobileNet_v3_Small	17.89	**46.28**	45.05	44.92	45.75
	ShuffleNet_v2_x0_5	23.97	49.21	49.50	48.50	**52.40**
T-2	VGG16	40.63	45.20	**53.15**	49.82	46.22
	VGG19	44.92	48.29	51.09	**52.44**	49.53
	ResNet18	38.83	41.87	42.02	42.62	**42.73**
	AlexNet	42.32	**45.03**	44.20	43.91	43.44
	GoogLeNet	39.67	42.14	42.09	42.46	**43.66**
	DenseNet121	42.20	**43.36**	43.15	43.35	42.75
	EfficientNet_b0	40.18	40.18	38.89	**40.97**	40.91
	MnasNet1_0	39.31	59.94	60.15	52.08	**61.31**
	MobileNet_V2	41.07	41.36	43.84	**44.75**	44.18
	MobileNet_V3_Small	35.23	**45.05**	40.73	43.49	41.33
	ShuffleNet_V2_x0_5	27.87	**41.94**	41.42	41.35	41.46

images through GANs are then assorted with the original source domain images.

Specifically, we generate 200, 300, and 500 samples per class from random noise. We compared the performance of this assorted dataset with the target domain in two scenarios: without fine-tuning the pre-trained model and with fine-tuning the pre-trained model.

3) ZsTL WITH DATA AUGMENTATION USING CROPPING (ZsTL-DAC)

In this experiment setup, we focused to generate new images from original images. The original images from different classes will taken to perform data augmentation. In this process, initially we take original image rotated with specific angle and saved the generated image. The new generated image corners may be out of dimension then edges cropped to fit the fixed dimension. The angles considered for rotation is starting with 5, multiples of 5 up to 90 degrees angle.

4) ZsTL WITH DISCRIMINATIVE INFORMATION PRESERVATION (ZsTL-DIP)

To preserve the discriminative information during the experiment, we incorporate center-based losses(ZsTL-DIP-CL) and triplet-based losses(ZsTL-DIP-TL). These additional loss functions are applied during the fine-tuning process of the existing pre-trained model, alongside the cross-entropy loss.

5) ZsTL WITH DATA AUGMENTATION USING GANs AND DISCRIMINATIVE INFORMATION PRESERVATION (ZsTL-DAG+DIP)

In this context, we enhance the data from the source domain by incorporating images generated from CGANs, while simultaneously preserving the essential characteristics of the

target domain through the utilization of center and triplet losses.

6) ZsTL WITH DATA AUGMENTATION USING CROPPING AND DISCRIMINATIVE INFORMATION PRESERVATION (ZsTL-DAC+DIP)

In this experiment, we enhance the data from the source domain by employing simple techniques such as rotation, flipping, and so on. The augmented data is then fine-tuned using discriminative losses to further refine its quality and effectiveness.

7) ZsTL WITH DATA AUGMENTATION USING CROPPING AND GANs, DISCRIMINATIVE INFORMATION PRESERVATION (ZsTL-DAC+DAG+DIP)

In this experiment, we incorporate both data generated using GANs models and data obtained through cropping techniques. Subsequently, the augmented data is fine-tuned using discriminative losses, which serve to enhance its quality and effectiveness.

D. RESULTS AND DISCUSSION
1) ZERO SHORT TRANSFER LEARNING (ZsTL)
The first column in Table 5 displays the testing results of the models on the target domain, where the models were not trained using any data from the source domain. Models trained with additional data generally demonstrate higher accuracy compared to those without additional training. This indicates the importance of training on relevant datasets to improve model performance. Among the models considered, the VGG16 pre-trained model demonstrates the best performance with the highest accuracy of 50.27%.

TABLE 7. Zero short transfer learning with discriminative information preservation.

TASK	Models	Original Mode		ZsTL-DIP	
		Without-training	With-training	Center Loss	Triplet Loss
T-1	VGG16	50.27	**71.46**	69.56	70.77
	VGG19	49.95	65.42	67.66	**71.27**
	ResNet18	42.70	52.23	**59.99**	59.53
	AlexNet	49.21	**56.22**	51.02	46.47
	GoogLeNet	31.22	54.41	**57.52**	55.53
	DenseNet121	42.33	49.30	**62.50**	58.46
	EfficientNet_b0	36.33	66.68	69.19	**73.28**
	MnasNet1_0	09.61	18.16	**27.04**	21.28
	MobileNet_v2	45.07	61.43	64.91	**71.00**
	MobileNet_v3_Small	17.89	46.28	44.37	**46.70**
	ShuffleNet_V2_x0_5	23.97	49.21	53.20	**57.62**
T-2	VGG16	40.63	45.20	44.06	**47.38**
	VGG19	44.92	48.29	**53.64**	45.76
	ResNet18	38.83	41.87	39.44	**42.45**
	AlexNet	42.32	**45.03**	41.56	40.83
	GoogLeNet	39.67	42.14	**42.56**	42.43
	DenseNet121	42.20	**43.36**	43.09	41.12
	EfficientNet_b0	40.18	40.18	38.83	**43.05**
	MnasNet1_0	39.31	**59.94**	57.61	45.63
	MobileNet_V2	41.07	41.36	**41.45**	40.73
	MobileNet_v3_Small	35.23	**45.05**	42.54	41.69
	ShuffleNet_v2_x0_5	27.87	**41.94**	41.76	40.16

TABLE 8. Zero short transfer learning with discriminator on GANs model.

TASK	Model	ZsTL- DIP (Center Loss- CeL, Triplet Loss- TrL)+ DAG-No.of instances per class							
		CeL	CeL+AUG-200	CeL+AUG-300	CeL+AUG-500	TrL	TrL+AUG-200	TrL+AUG-300	TrL+AUG-500
T-1	VGG16	69.56	68.96	67.14	69.63	70.77	69.06	66.13	68.74
	VGG19	67.66	55.23	53.16	60.93	71.27	69.74	75.66	**75.70**
	ResNet18	59.99	58.41	56.51	55.25	59.53	60.22	64.40	**69.79**
	AlexNet	51.02	52.60	49.86	50.55	46.47	48.60	46.46	48.97
	GoogLeNet	57.52	55.02	57.95	58.41	55.53	56.97	65.00	**70.39**
	DenseNet121	62.05	58.78	60.87	60.13	58.46	56.31	**63.42**	63.19
	EfficientNet_b0	69.19	69.01	67.33	70.91	**73.28**	65.52	67.89	59.80
	MnasNet1_0	27.04	27.00	31.74	33.87	21.28	45.21	**46.79**	41.03
	MobileNet_v2	64.91	69.42	64.92	66.21	**71.00**	68.26	68.68	66.86
	MobileNet_V3_Small	44.37	45.63	47.12	46.09	46.70	**48.51**	47.25	43.95
	ShuffleNet_V2_x0_5	53.20	52.18	53.53	49.81	**57.62**	55.52	54.73	52.55
T-2	VGG16	44.06	41.36	42.86	40.94	**47.38**	43.42	43.04	44.33
	VGG19	**53.64**	44.10	43.82	40.56	45.76	44.91	47.66	42.51
	ResNet18	39.44	43.65	**45.18**	43.45	42.45	42.38	41.14	42.14
	AlexNet	41.56	41.69	41.80	41.38	40.83	42.43	39.34	41.54
	GoogLeNet	42.56	42.56	42.18	**43.12**	42.43	42.36	42.80	43.05
	DenseNet121	43.09	44.36	43.76	**44.87**	41.12	40.80	41.69	41.94
	EfficientNet_b0	38.83	42.72	47.63	43.43	43.05	41.74	41.14	**50.96**
	MnasNet1_0	57.61	48.01	52.34	49.09	45.63	43.14	42.89	43.32
	MobileNet_V2	41.45	41.69	43.47	**43.74**	40.73	43.47	41.87	41.67
	MobileNet_V3_Small	42.54	44.29	42.76	41.72	41.69	**46.43**	41.78	42.20
	ShuffleNet_v2_x0_5	41.76	41.65	41.69	**43.96**	40.16	39.05	41.74	42.14

Following closely is the VGG19 model, achieving an accuracy of 49.95%. Among the more computationally efficient models, MobileNet-v2 performs reasonably well with an accuracy of 45.07%, surpassing the other models. However, the MnaseNet1_0 model lags behind and exhibits poor performance compared to all the other models. In the second column of Table 5, the testing results of the models on the target domain are presented, where the models were trained using data from the source domain containing classes unknown to the target domain. The improvement in all tasks clearly demonstrates that testing accuracy can be enhanced by leveraging data from other datasets with unknown classes. Among all the models, the VGG16 model achieves the highest accuracy of 71.46%, closely followed by the EfficientNet_b0 and VGG19 models with accuracies of 66.68% and 65.42% respectively. Notably, the EfficientNet_b0 model surprises with an impressive 83.53% accuracy improvement. Comparing the models in terms of the percentage improvement in accuracy, the MobileNet-V3-Small model stands out with the highest improvement of 158%, making it the most significant improvement among all the compared models. Following closely is the ShuffleNet_v2_x05 model, which exhibits an improvement of 105%.

TABLE 9. Zero short transfer learning with discriminator on augmented images.

TASK	Model	ZsTL- DIP (Center Loss- CeL/ Triplet Loss- TrL) +DAC -No.of instances per class							
		CeL	CeL+DAC-200	CeL+DAC-300	CeL+DAC-500	TrL	TrL+DAC-200	TrL+DAC-300	TrL+DAC-500
T-1	VGG16	69.56	72.06	66.64	72.81	70.77	73.23	**78.34**	74.76
	VGG19	67.66	64.75	68.55	65.63	71.27	**77.40**	70.93	74.31
	ResNet18	59.99	52.83	58.92	60.40	59.53	**69.28**	66.49	67.24
	AlexNet	51.02	50.83	52.83	49.34	46.47	**56.69**	46.46	46.46
	GoogLeNet	57.52	57.52	57.10	61.94	55.53	55.11	57.57	**66.44**
	DenseNet121	62.05	64.73	65.24	59.43	58.46	67.33	67.10	**62.73**
	EfficientNet_b0	69.19	68.02	66.31	68.49	73.28	64.22	69.61	67.57
	MnasNet1_0	27.04	28.90	37.96	37.22	21.28	36.29	**47.39**	47.35
	MobileNet_V2	64.91	62.73	63.38	62.40	**71.00**	67.89	60.26	53.94
	Mobilenet_V3_Small	44.37	43.26	45.21	46.60	46.70	49.02	**50.88**	47.53
	ShuffleNet_v2_x0_5	53.20	51.16	50.74	48.74	57.62	54.87	56.18	**59.20**
T-2	VGG16	44.06	42.54	43.88	47.74	47.38	49.15	47.69	**50.53**
	VGG19	53.64	50.66	44.62	47.30	45.76	49.00	**58.65**	54.93
	ResNet18	39.44	44.92	41.27	43.92	42.45	42.74	42.89	**45.34**
	AlexNet	41.56	41.83	42.00	41.83	40.83	41.34	42.43	41.92
	GoogLeNet	**42.56**	42.56	42.03	42.52	42.43	42.32	42.52	42.43
	DenseNet121	43.09	43.12	42.69	42.65	41.12	**43.89**	41.85	42.89
	EfficientNet_b0	38.83	38.27	38.49	40.63	**43.05**	42.03	39.45	42.14
	MnasNet1_0	57.61	**67.05**	65.21	53.10	45.63	49.03	52.10	49.25
	MobileNet_V2	41.45	**47.01**	45.12	44.05	40.73	45.01	44.63	46.32
	MobileNet_V3_Small	42.54	42.83	43.03	43.23	41.69	40.85	42.43	41.79
	ShuffleNet_V2_x0_5	41.76	41.78	41.40	41.92	40.16	40.89	40.40	40.80

Without fine-tuning the pre-trained models, VGG16 achieves the highest accuracy of 44.92%, surpassing all others. Once the pre-trained model parameters are refined, MnasNet1.0 stands out with an impressive accuracy of 59.73%, marking a substantial leap from its initial 39.31% accuracy.

In summary, the findings underscore the potential benefits of leveraging data from diverse datasets to improve the testing accuracy of models in domains with unknown classes.

2) ZsTL WITH DATA AUGMENTATION USING GANs

In augmentation using GANs models, a fascinating pattern emerges when adding different numbers of images per class. The VGG16 model stands out, surpassing all other models with an impressive accuracy of 73.14%. Augmenting 200 images per class results in a 2% improvement for the VGG16 model, a 5% improvement for the VGG19 model, and a significant 9% improvement for the AlexNet model, compared to their respective baseline accuracies. Furthermore, other models also demonstrate improvements when augmenting with varying numbers of images per class. The ResNet18 model shows an 8% improvement when adding 300 images per class, the DenseNet121 model achieves a 12% improvement under the same augmentation, and the ShuffleNet model experiences a 4% improvement. Likewise, when augmenting with 500 images per class, the GoogLeNet, EfficientNet, and MobileNet-v2 models show improvements of 4%, 7%, and 2% respectively. These results underscore the efficacy of augmentation using GANs in enhancing model performance across various architectures. The VGG16 model, in particular, shines as it achieves the highest accuracy compared to all other models in this specific scenario. This highlights the potential of GANs-based augmentation as a valuable technique for improving the

performance of deep learning models in classification tasks. Upon augmenting the synthetic data set with 200 images per class, VGG16, VGG19, ResNet18, and MobileNet-V3-Small emerge as the top performers in terms of accuracy amongst the various models. Notably, VGG19 exhibits a noteworthy improvement, boasting a 23% increase in accuracy. However, introducing 300 and 500 instances per class into the source domain may not yield favorable outcomes for most of the fine-tuned models. Nonetheless, there are improvements observed in the GoogLeNet and EfficientNet_b0 models after the addition of 500 instances per class in the source domain. Likewise, the MobileNet-V2 model achieves its peak accuracy after the introduction of 300 instances per class.

In summary, the findings suggest that training models with additional relevant data can improve accuracy. Different models show varying levels of performance based on the tasks and training data used. Fine-tuning and data augmentation can also have significant impacts on model accuracy.

3) ZsTL WITH DATA AUGMENTATION USING CROPPING

When applying rotation and cropping techniques for augmentation, intriguing insights emerge when considering varying quantities of images per class, as depicted in Table 6. Notably impressive accuracy results are attained through the utilization of rotation and cropping methods, especially when maintaining a uniform number of images across classes.

In the context of Task-1 accuracy evaluations, in relation to the trained model's performance, augmenting with 500 instances per class yields the highest accuracy at 75.38%, marking a substantial enhancement of 3.92%. Likewise, considerable accuracy boosts of 4.17%, 1.56%, 4%, 4.85%, 3.19%, and 4% are realized for the VGG16, VGG19, GoogLeNet, DenseNet121, and ShuffleNet_v2_x0_5 models, respectively.

TABLE 10. Zero short transfer learning with discriminator on GAN, augmented images.

TASK	Model	ZsTL- DIP (Center Loss- CeL, Triplet Loss- TrL)+ DAC+DAG-No.of instances per class							
		CeL	CeL+AUG+ AUC-200	CeL+AUG+ AUC-300	CeL+AUG+ AUC-500	TrL	TrL+AUG+ AUC-200	TrL+AUG+ AUC-300	TrL+AUG+ AUC-500
T-1	VGG16	69.56	**72.17**	70.78	68.69	70.77	67.65	67.24	68.10
	VGG19	67.66	68.80	67.38	63.04	71.27	73.90	**75.17**	66.55
	ResNet18	59.99	60.26	57.06	57.76	59.53	**63.75**	61.52	54.50
	AlexNet	51.02	48.18	51.67	47.63	46.47	46.46	46.46	51.53
	GoogLeNet	57.52	60.36	57.76	59.71	55.53	63.42	**69.60**	65.52
	DenseNet121	62.05	**71.70**	60.64	61.80	58.46	61.47	64.17	62.45
	EfficientNet_b0	69.19	66.44	66.35	68.83	**73.28**	62.96	58.87	62.54
	MnasNet1_0	27.04	36.47	34.43	46.00	21.28	45.53	43.63	**47.90**
	MobileNet_v2	64.91	61.24	61.80	64.96	71.00	**72.30**	60.40	68.21
	MobileNet_V3_Small	44.37	42.65	43.02	45.86	46.70	45.77	**50.41**	48.28
	ShuffleNet_v2_x0_5	53.20	55.06	52.09	49.44	**57.62**	54.83	55.06	54.13
T-2	VGG16	44.06	45.46	47.38	45.26	**47.38**	43.37	46.18	44.86
	VGG19	**53.64**	50.68	51.36	48.50	45.76	42.82	46.95	40.81
	ResNet18	39.44	43.23	44.43	**45.43**	42.45	41.72	44.27	44.03
	AlexNet	41.56	41.54	41.40	41.36	40.83	38.76	40.89	39.76
	GoogLeNet	42.56	42.49	42.52	**42.58**	42.43	42.52	42.47	42.43
	DenseNet121	43.09	43.03	42.18	**45.01**	41.12	40.96	42.38	41.32
	EfficientNet_b0	38.83	47.32	44.74	**48.65**	43.05	46.58	46.92	45.54
	MnasNet1_0	57.61	**63.32**	54.58	58.19	45.63	42.98	52.16	44.78
	MobileNet_v2	41.45	45.03	47.65	**50.34**	40.73	47.96	43.09	45.52
	MobileNet_v3_small	42.54	42.09	44.09	40.07	41.69	45.43	**47.07**	41.92
	ShuffleNet_V2_x0_5	41.76	41.67	**42.34**	41.29	40.16	40.25	39.69	41.12

In Task-2's accuracy assessment, the most noteworthy progress is observed in the VGG16 model, achieving an impressive 53.15% accuracy increase when supplemented with 200 instances, compared to the base model's accuracy. The EfficientNet_b0 and MobileNet_v2 models exhibit accuracies of 40.97% and 44.75%, respectively, upon augmentation with 300 instances. Furthermore, employing 500 instances results in accuracies of 42.73%, 43.66%, and 61.31% for the ResNet18, GoogLeNet, and MnasNet_0 models, respectively.

The inclusion of rotation and cropping techniques for data augmentation leads to substantial accuracy improvements, particularly when maintaining a consistent number of augmented images per class, across a variety of model architectures.

4) ZsTL WITH DISCRIMINATIVE INFORMATION PRESERVATION (ZsTL-DIP)

The experimental results employing two distinct discriminators, namely Center Loss and Triplet Loss, are presented in Table 7. Regarding T-1, when employing the Triplet Loss discriminator, the achieved highest accuracies are as follows: VGG 19 attains 71.27%, AlexNet achieves 73.28%, EffientNet_b0 reaches 71%, MobileNet_v2 scores 46.70%, and both MobileNet_V3_Small and ShuffleNet_v2_x0_5 achieve 57.62%. On the other hand, utilization of the Center Loss discriminator yields significant accuracies of 59.99%, 57.52%, 62.05%, and 27.04% for ReseNet 18, GoogLeNet, DenseNet121, and MnasNet1_0 respectively. Notably, the accuracy of the MnasNet1_0 model experiences a remarkable increase of 181.37%.

In the context of T-2, the Center Loss discriminator demonstrates highest accuracies of 53.64%, 42.56%, and 41.45% for VGG 19, GoogLeNet, and MobileNet_V2 respectively. Meanwhile, the Triplet Loss showcases its effectiveness with models VGG 16, ResNet18, and EfficientNet_b0 achieving accuracies of 47.38%, 42.45%, and 43.05% respectively.

Upon comparing the performance of Triplet Loss and Center Loss, a discernible pattern emerges: Triplet Loss consistently outperforms Center Loss across the majority of the models. These comparative results underscore the significance of maintaining the discriminative information of source domain samples, which notably enhances the performance of the target domain model.

5) ZsTL WITH DAG+DIP

Table 8 presents a performance evaluation of various models and configurations, focusing on the achieved accuracy when using data instances generated through a GAN with a discriminator. In the context of Task T-1, a thorough examination of the data in Table 8 reveals a distinct pattern: when employing a GAN model to augment images, the triplet loss method effectively emphasizes the preservation of discriminative information compared to the center loss method.

For instance, notable models such as vGG19, ResNet18, and GoogLeNet consistently outperform other settings, especially when combined with the TrL+AUG-500 configuration. Similarly, the MobilevNet2 model demonstrates strong performance when coupled with the TrL+AUG-200 setting. These observations emphasize the superiority of the triplet loss approach in maintaining discriminative information while augmenting data through a GAN.

When delving into task T-2, an interesting contrast emerges: the utilization of center loss to preserve discriminative information surpasses the performance of triplet loss. To illustrate, when combined with the CeL+AUG-500 configuration, specific models such as GoogLeNet, DenseNet_b0, MobileNet_v2, and ShuffleNet_v2 × 0_5 consistently excel, outperforming all other settings.

In summary, it becomes evident that the preservation of discriminative information, coupled with augmentation techniques, plays a crucial role in enhancing the performance of the source domain model as it translates to the target domain model.

6) ZsTL WITH DAC+DIP

Table 9 provides an insightful analysis of various models and configurations based on the accuracy achieved when using Data Augmentation through Cropping and Rotation in conjunction with a discriminator. Several models consistently demonstrate strong performance across different configurations, notably VGG 16 and VGG 19, while others exhibit more variability, such as MnasNet1_0 and MobileNet_v3_Small.

The experiment involves instances generated through Data Augmentation, with the addition of 200, 300, and 500 instances per class. Both Center Loss and Triplet Loss are applied to Task T-1 and T-2. In T-1, employing Triplet Loss with 300 instances per class leads to remarkable accuracy improvements, with MnasNet1_0 experiencing an astonishing 160.95% increase to 47.39%, followed by DenseNet121 with a 36.57% rise to 62.73%. VGG 16 achieves an accuracy of 78.34% with 300 instances and triplet loss, while GoogLeNet reaches 70.39%, and DenseNet performs at 63.42%.

Moving to T-2, a notable surge in accuracy is observed. VGG 19, utilizing triplet loss with 300 instances per class added, achieves an outstanding 21.45% increase in accuracy. Additionally, MobileNet_v2, with center loss on 200 instances per class added, achieves a remarkable accuracy improvement of 13.66%. These findings underscore the substantial impact of employing Data Augmentation with Cropping and Rotation, in tandem with discriminative loss methods, on model performance.

7) ZsTL WITH DAG+DAC+DIP

Table 10 provides a comprehensive overview of accuracy across different models and configurations, focusing on instances generated through Data Augmentation, synthetic instances, and original instances combined with Center Loss and Triplet Loss discriminators. The experiment involves the incorporation of augmented instances alongside original and synthetic instances, with the addition of 200, 300, and 500 instances per class. Both Center Loss and Triplet Loss are employed for Task T-1 and T-2.

As anticipated, many models demonstrate superior performance compared to other experimental settings. In T-1, the utilization of Triplet Loss with 500 instances per class

results in the most notable accuracy increase. For instance, MnasNet1_0 experiences a remarkable 76.99% accuracy rise to 47.90%, while DenseNet121 achieves a commendable 71.7% accuracy with Center Loss and 200 instances per class. Among the models, VGG 19 stands out, achieving an elite accuracy of 75.17% with Triplet Loss and 300 instances.

Transitioning to T-2, remarkable accuracy enhancements are observed. MobileNet_v2, DenseNet121, GoogLeNet, and ShuffleNet_V2_×0_5 models achieve an outstanding 5.74%, 3.4%, 2.3% and 4.5% increase in accuracy with Center Loss and 500 instances per class. Additionally, EfficientNet_b0 attains the highest accuracy across all settings, with a notable 50.96% accuracy on Triplet Loss and 500 instances per class.

These findings underscore the substantial impact of combining various instance sources with discriminative loss methods, further validating their effectiveness in enhancing model performance.

E. CONVERGES ANALYSIS

To gain insights into the convergence patterns exhibited by various models across different scenarios, we utilized visualizations to depict the training and testing accuracies in Figure 6. The graphs present the Training and Testing accuracies on the Y-axis and the number of epochs on the X-axis. This approach allows us to track the evolution of accuracy values over time, potentially unveiling trends and convergence behaviors. These visual representations hold the potential to yield valuable insights into the models' learning processes and their performances under diverse conditions.

Graph (a) illustrates the outcomes of Zero Short Transfer Learning with Center Loss and Triplet Loss Discriminators executed on Data Augmented images generated by GANs with 200, 300, and 500 instances per class using the EfficientNet_b0 model. The graph demonstrates that Training accuracies remain consistently high regardless of the instance count. Concerning Testing accuracies, the results for 500 instances per class surpass those of 200 and 300 instances. The accuracies of 200 and 300 instances exhibit minor variations, with slightly higher values for the latter.

Graph (b) showcases the results of Zero Short Transfer Learning with Center Loss and Triplet Loss Discriminators executed on Data Augmented images generated by Cropping, featuring 200, 300, and 500 instances per class using the VGG 19 model. The graph indicates that accuracy values fluctuate until around 50 epochs, after which variations stabilize. Notably, the Testing accuracy for 500 instances per class starts notably higher compared to 200 and 300 instances.

Graph (c) displays the outcomes of Zero Short Transfer Learning with Center Loss and Triplet Loss Discriminators executed on the MobileNet_V2 model. The graph illustrates that Testing accuracy values exhibit more fluctuations with Triplet Loss compared to Center Loss. With Central Loss, Training accuracy rises until approximately 20 epochs, beyond which it gradually and consistently stabilizes. In contrast, the accuracy of Triplet Loss fluctuates throughout all
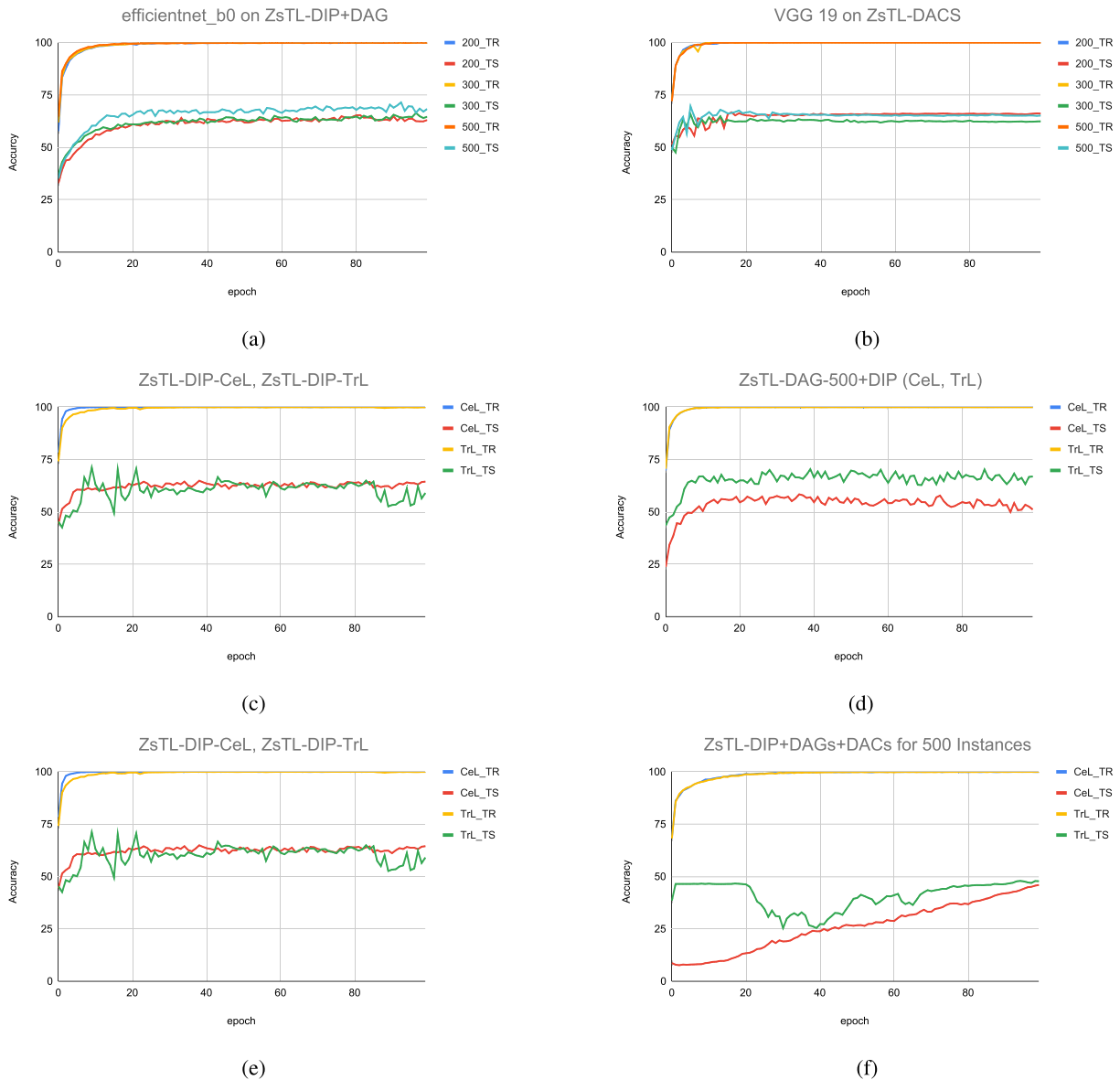
FIGURE 6. Illustration of accuracy of diffrent models with instances generated by GANs model, Augmented models with center Loss, triplet Loss. In graph symbols notation, TR- Train Accuracy, TS- Test Accuracy, CeL - Center Loss, TrL - Triplet Loss.

epochs due to the dynamic selection of anchor points and classes.

Graph (d) portrays the outcomes of Zero Short Transfer Learning with Center Loss and Triplet Loss Discriminators executed on Data Augmented images generated by GANs with 500 instances per class using the GoogLeNet model. The graph shows that Training accuracies remain uniformly high regardless of instance count. Regarding testing accuracies, Triplet Loss yields higher values compared to Center Loss.

Graph (e) presents the results of Zero Short Transfer Learning with Center Loss and Triplet Loss Discriminators executed on Data Augmented images generated by cropping, involving 500 instances per class using the GoogLeNet model. Similar to the previous scenario, Training accuracies

exhibit uniformity across instances. Testing accuracies for both Triplet Loss and Center Loss are nearly identical, with Triplet Loss having slightly higher values.

Graph (f) depicts the results of Zero Short Transfer Learning with Center Loss and Triplet Loss Discriminators executed on Data Augmented images generated by GANs and cropping, employing 500 instances per class with the MnasNet1_0 model. The graph highlights that training accuracies for both Triplet Loss and Center Loss are consistently equivalent. However, testing accuracies differ. The initial testing accuracy for Triplet Loss is higher, gradually decreasing and then rising again as epochs progress. On the contrary, the testing accuracy for Center Loss starts lower but steadily improves as epochs increase. By the 100th epoch, the

testing accuracy for Center Loss lags behind Triplet Loss by 1.07%.

V. CONCLUSION

Our paper introduces a Zero-shot Transfer Learning framework as a solution to the challenge of an inadequate data availability in the target domain for leaf Disease Classification. By integrating cutting-edge CNN models and a range of techniques such as data augmentation, synthetic data generation, and robust discriminative losses, we establish a robust methodology for effectively transferring knowledge from a well-populated source domain to a less abundant target domain. Through comprehensive experimentation on plant disease classification datasets, we validate the potency of each model and technique within the framework.

The results obtained from our experiments highlight the significant improvements in classification accuracy achieved through the implementation of these techniques. This not only confirms the effectiveness of the Zero-shot Transfer Learning framework but also underscores the broader potential of this approach in enhancing disease classification within agricultural contexts. As agriculture continues to face challenges in a rapidly changing environment, leveraging innovative methods like Zero-shot Transfer Learning could play a crucial role in optimizing crop yield which ensures food security. Our research contributes to the ongoing advancements in agricultural technology and opens up avenues for further exploration in the realm of data-driven approaches for tackling real-world agricultural challenges.

DECLARATION OF THE INTEREST

The authors claim no financial or other conflicts of interest that could have influenced the study's findings.

REFERENCES

[1] G. Tilahun, P. Natarajan, A. Getahun, and T. Teame, "Bibliography of aquatic sciences, fisheries and aquaculture in Ethiopia: Bibliography," *East Afr. J. Biophysical Comput. Sci.*, pp. 1–61, 2021.

[2] L. Li, "Commodity prices volatility and economic growth: Empirical evidence from natural resources industries of China," *Resour. Policy*, vol. 80, Jan. 2023, Art. no. 103152.

[3] A. Raihan, M. I. Pavel, D. A. Muhtasim, S. Farhana, O. Faruk, and A. Paul, "The role of renewable energy use, technological innovation, and forest cover toward green development: Evidence from Indonesia," *Innov. Green Develop.*, vol. 2, no. 1, Mar. 2023, Art. no. 100035.

[4] W. Zhao, W. Yamada, T. Li, M. Digman, and T. Runge, "Augmenting crop detection for precision agriculture with deep visual transfer learning—A case study of bale detection," *Remote Sens.*, vol. 13, no. 1, p. 23, Dec. 2020.

[5] J. G. A. Barbedo, "An automatic method to detect and measure leaf disease symptoms using digital image processing," *Plant Disease*, vol. 98, no. 12, pp. 1709–1716, Dec. 2014.

[6] F. Qin, D. Liu, B. Sun, L. Ruan, Z. Ma, and H. Wang, "Identification of alfalfa leaf diseases using image recognition technology," *PLoS ONE*, vol. 11, no. 12, Dec. 2016, Art. no. e0168274.

[7] Z. Ramedani, M. Omid, A. Keyhani, S. Shamshirband, and B. Khoshnevisan, "Potential of radial basis function based support vector regression for global solar radiation prediction," *Renew. Sustain. Energy Rev.*, vol. 39, pp. 1005–1011, Nov. 2014.

[8] J. G. A. Barbedo, "A new automatic method for disease symptom segmentation in digital photographs of plant leaves," *Eur. J. Plant Pathol.*, vol. 147, no. 2, pp. 349–364, Feb. 2017.

[9] E. Hossain, M. F. Hossain, and M. A. Rahaman, "A color and texture based approach for the detection and classification of plant leaf disease using KNN classifier," in *Proc. Int. Conf. Electr., Comput. Commun. Eng. (ECCE)*, Feb. 2019, pp. 1–6.

[10] R. M. Mohana, C. K. K. Reddy, P. R. Anisha, and B. V. R. Murthy, *WITHDRAWN: Random Forest Algorithms for the Classification of Tree-Based Ensemble*. Amsterdam, The Netherlands: Elsevier, 2021.

[11] T. Rumpf, A.-K. Mahlein, U. Steiner, E.-C. Oerke, H.-W. Dehne, and L. Plümer, "Early detection and classification of plant diseases with support vector machines based on hyperspectral reflectance," *Comput. Electron. Agricult.*, vol. 74, no. 1, pp. 91–99, Oct. 2010.

[12] D. Al Bashish, M. Braik, and S. Bani-Ahmad, "Detection and classification of leaf diseases using K-means-based segmentation and," *Inf. Technol. J.*, vol. 10, no. 2, pp. 267–275, 2011.

[13] J. Gu, "Recent advances in convolutional neural networks," *Pattern Recognit.*, vol. 77, pp. 354–377, May 2018.

[14] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *Proc. Int. Conf. Eng. Technol. (ICET)*, Aug. 2017, pp. 1–6.

[15] F. Liu, G. Zhang, and J. Lu, "Heterogeneous domain adaptation: An unsupervised approach," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 12, pp. 5588–5602, Dec. 2020.

[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014.

[17] X. Fan, P. Luo, Y. Mu, R. Zhou, T. Tjahjadi, and Y. Ren, "Leaf image based plant disease identification using transfer learning and feature fusion," *Comput. Electron. Agricult.*, vol. 196, May 2022, Art. no. 106892.

[18] M. Long, J. Wang, G. Ding, S. J. Pan, and P. S. Yu, "Adaptation regularization: A general framework for transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 5, pp. 1076–1089, May 2014.

[19] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 199–210, Feb. 2011.

[20] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer joint matching for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1410–1417.

[21] J. Wang, W. Feng, Y. Chen, H. Yu, M. Huang, and P. S. Yu, "Visual domain adaptation with manifold embedded distribution alignment," 2018, *arXiv:1807.07258*.

[22] J. Zhang, W. Li, and P. Ogunbona, "Transfer learning for cross-dataset recognition: A survey," 2017, *arXiv:1705.04396*.

[23] A. Farahani, S. Voghoei, K. Rasheed, and H. R. Arabnia, "A brief review of domain adaptation," in *Advances in Data Science and Information Engineering*. Cham, Switzerland: Springer, 2021, pp. 877–894.

[24] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, Oct. 2018.

[25] J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue, and G. Zhang, "Transfer learning using computational intelligence: A survey," *Knowl.-Based Syst.*, vol. 80, pp. 14–23, May 2015.

[26] R. K. Sanodiya, J. Mathew, S. Saha, and M. D. Thalakottur, "A new transfer learning algorithm in semi-supervised setting," *IEEE Access*, vol. 7, pp. 42956–42967, 2019.

[27] R. K. Sanodiya and L. Yao, "A subspace based transfer joint matching with Laplacian regularization for visual domain adaptation," *Sensors*, vol. 20, no. 16, p. 4367, Aug. 2020.

[28] J. Liu, B. Kuipers, and S. Savarese, "Recognizing human actions by attributes," in *Proc. CVPR*, Jun. 2011, pp. 3337–3344.

[29] C. H. Lampert, H. Nickisch, and S. Harmeling, "Attribute-based classification for zero-shot visual object categorization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 3, pp. 453–465, Mar. 2014.

[30] T. Mensink, E. Gavves, and C. G. M. Snoek, "COSTA: Co-occurrence statistics for zero-shot classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2441–2448.

[31] D. Jayaraman and K. Grauman, "Zero-shot recognition with unreliable attributes," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 3464–3472.

[32] K. Tanwisuth, "A prototype-oriented framework for unsupervised domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 17194–17208.

[33] W. Yang, C. Yang, S. Huang, L. Wang, and M. Yang, "Few-shot unsupervised domain adaptation via meta learning," in *Proc. IEEE Int. Conf. Multimedia Expo. (ICME)*, Jul. 2022, pp. 1–6.

[34] A. Frome, "Devise: A deep visual-semantic embedding model," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 26, 2013, pp. 2121–2129.

[35] Y. Fu, T. M. Hospedales, T. Xiang, Z. Fu, and S. Gong, "Transductive multi-view embedding for zero-shot recognition and annotation," in *Proc. Eur. Conf. Comput. Vis.*, Zurich, Switzerland, Sep. 2014, pp. 584–599.

[36] Z. Fu, T. A. Xiang, E. Kodirov, and S. Gong, "Zero-shot object recognition by semantic manifold distance," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2635–2644.

[37] G. Dhingra, V. Kumar, and H. D. Joshi, "Study of digital image processing techniques for leaf disease detection and classification," *Multimedia Tools Appl.*, vol. 77, no. 15, pp. 19951–20000, Aug. 2018.

[38] V. Singh and A. K. Misra, "Detection of plant leaf diseases using image segmentation and soft computing techniques," *Inf. Process. Agricult.*, vol. 4, no. 1, pp. 41–49, Mar. 2017.

[39] A. Chlingaryan, S. Sukkarieh, and B. Whelan, "Machine learning approaches for crop yield prediction and nitrogen status estimation in precision agriculture: A review," *Comput. Electron. Agricult.*, vol. 151, pp. 61–69, Aug. 2018.

[40] R. Pydipati, T. F. Burks, and W. S. Lee, "Identification of citrus disease using color texture features and discriminant analysis," *Comput. Electron. Agricult.*, vol. 52, nos. 1–2, pp. 49–59, Jun. 2006.

[41] A. Camargo and J. S. Smith, "An image-processing based algorithm to automatically identify plant disease visual symptoms," *Biosyst. Eng.*, vol. 102, no. 1, pp. 9–21, Jan. 2009.

[42] Q. He, B. Ma, D. Qu, Q. Zhang, X. Hou, and J. Zhao, "Cotton pests and diseases detection based on image processing," *TELKOMNIKA Indonesian J. Electr. Eng.*, vol. 11, no. 6, pp. 3445–3450, Jun. 2013.

[43] C. C. Tucker and S. Chakraborty, "Quantitative assessment of lesion characteristics and disease severity using digital image processing," *J. Phytopathol.*, vol. 145, no. 7, pp. 273–278, Jul. 1997.

[44] J. R. Aduwo, E. Mwebaze, and J. A. Quinn, "Automated vision-based diagnosis of cassava mosaic disease," in *Proc. ICDM (Workshops)*, 2010, pp. 114–122.

[45] K. Muthukannan and P. Latha, "A PSO model for disease pattern detection on leaf surfaces," *Image Anal. Stereol.*, vol. 34, no. 3, p. 209, Jul. 2015.

[46] Z. Jian and Z. Wei, "Support vector machine for recognition of cucumber leaf diseases," in *Proc. 2nd Int. Conf. Adv. Comput. Control*, vol. 5, Mar. 2010, pp. 264–266.

[47] M. A. Chandra and S. S. Bedi, "Survey on SVM and their application in image classification," *Int. J. Inf. Technol.*, vol. 13, no. 5, pp. 1–11, Oct. 2021.

[48] M. Tanveer, T. Rajani, R. Rastogi, Y. H. Shao, and M. A. Ganaie, "Comprehensive review on twin support vector machines," *Ann. Oper. Res.*, pp. 1–46, 2022.

[49] A. H. B. A. Wahab, R. Zahari, and T. H. Lim, "Detecting diseases in chilli plants using K-means segmented support vector machine," in *Proc. 3rd Int. Conf. Imag., Signal Process. Commun. (ICISPC)*, Jul. 2019, pp. 57–61.

[50] V. P. Kour and S. Arora, "Particle swarm optimization based support vector machine (P-SVM) for the segmentation and classification of plants," *IEEE Access*, vol. 7, pp. 29374–29385, 2019.

[51] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Frontiers Plant Sci.*, vol. 7, p. 1419, Sep. 2016.

[52] P. S. Thakur, P. Khanna, T. Sheorey, and A. Ojha, "Trends in vision-based machine learning techniques for plant disease identification: A systematic review," *Expert Syst. Appl.*, vol. 208, Dec. 2022, Art. no. 118117.

[53] S. M. Hassan and A. K. Maji, "Deep feature-based plant disease identification using machine learning classifier," *Innov. Syst. Softw. Eng.*, pp. 1–11, 2022.

[54] H. Lu, M. Du, K. Qian, X. He, and K. Wang, "GAN-based data augmentation strategy for sensor anomaly detection in industrial robots," *IEEE Sensors J.*, vol. 22, no. 18, pp. 17464–17474, Sep. 2022.

[55] J. A. Pandian, G. Geetharamani, and B. Annette, "Data augmentation on plant leaf disease image dataset using image manipulation and deep learning techniques," in *Proc. IEEE 9th Int. Conf. Adv. Comput. (IACC)*, Dec. 2019, pp. 199–204.

[56] C. Douarre, C. F. Crispim-Junior, A. Gelibert, L. Tougne, and D. Rousseau, "Novel data augmentation strategies to boost supervised segmentation of plant disease," *Comput. Electron. Agricult.*, vol. 165, Oct. 2019, Art. no. 104967.

[57] Q. Zeng, X. Ma, B. Cheng, E. Zhou, and W. Pang, "GANs-based data augmentation for citrus disease severity detection using deep learning," *IEEE Access*, vol. 8, pp. 172882–172891, 2020.

[58] D. Su, H. Kong, Y. Qiao, and S. Sukkarieh, "Data augmentation for deep learning based semantic segmentation and crop-weed classification in agricultural robotics," *Comput. Electron. Agricult.*, vol. 190, Nov. 2021, Art. no. 106418.

[59] T. Wei, J. Wang, W. Liu, H. Chen, and H. Shi, "Marginal center loss for deep remote sensing image scene classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 6, pp. 968–972, Jun. 2020.

[60] S. R. Singh, S. R. Dubey, S. Ms, S. Ventrapragada, and S. S. Dasharatha, "Joint triplet autoencoder for histopathological colon cancer nuclei retrieval," *Multimedia Tools Appl.*, pp. 1–20, 2023.

[61] I. Kavalerov, W. Czaja, and R. Chellappa, "cGANs with multi-hinge loss," 2019, *arXiv:1912.04216*.

[62] L. Liu, Q. Dou, H. Chen, J. Qin, and P.-A. Heng, "Multi-task deep model with margin ranking loss for lung nodule analysis," *IEEE Trans. Med. Imag.*, vol. 39, no. 3, pp. 718–728, Mar. 2020.

[63] P. Wang, K. Han, X.-S. Wei, L. Zhang, and L. Wang, "Contrastive learning based hybrid networks for long-tailed image classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 943–952.

[64] S. Zhang and C. Zhang, "Orthogonal locally discriminant projection for classification of plant leaf diseases," in *Proc. 9th Int. Conf. Comput. Intell. Secur.*, Dec. 2013, pp. 241–245.

[65] D. Argüeso, A. Picon, U. Irusta, A. Medela, M. G. San-Emeterio, A. Bereciartua, and A. Alvarez-Gila, "Few-shot learning approach for plant disease classification using images taken in the field," *Comput. Electron. Agricult.*, vol. 175, Aug. 2020, Art. no. 105542.

[66] H. Yin, Y. H. Gu, C.-J. Park, J.-H. Park, and S. J. Yoo, "Transfer learning-based search model for hot pepper diseases and pests," *Agriculture*, vol. 10, no. 10, p. 439, Sep. 2020.

[67] A. K. Rangarajan, R. Purushothaman, and A. Ramesh, "Tomato crop disease classification using pre-trained deep learning algorithm," *Proc. Comput. Sci.*, vol. 133, pp. 1040–1047, Jan. 2018.

[68] S. Nsumba, E. Mwebaze, E. Bagarukayo, and G. Maiga, "Automated image-based diagnosis of cowpea diseases," Agile, 2018.

[69] A. Ramcharan, K. Baranowski, P. McCloskey, B. Ahmed, J. Legg, and D. P. Hughes, "Deep learning for image-based cassava disease detection," *Frontiers Plant Sci.*, vol. 8, p. 1852, Oct. 2017.

[70] G. Wang, Y. Sun, and J. Wang, "Automatic image-based plant disease severity estimation using deep learning," *Comput. Intell. Neurosci.*, vol. 2017, pp. 1–8, Jul. 2017.

[71] C. Llorca, M. E. Yares, and C. Maderazo, "Image-based pest and disease recognition of tomato plants using a convolutional neural network," in *Proc. Int. Conf. Technol. Challenges Better World*, 2018.

[72] S. Vallabhajosyula, V. Sistla, and V. K. K. Kolli, "Transfer learning-based deep ensemble neural network for plant leaf disease detection," *J. Plant Diseases Protection*, vol. 129, no. 3, pp. 545–558, Jun. 2022.

[73] R. Zhao, Y. Zhu, and Y. Li, "CLA: A self-supervised contrastive learning method for leaf disease identification with domain adaptation," *Comput. Electron. Agricult.*, vol. 211, Aug. 2023, Art. no. 107967.

[74] Y. Ma, Z. Zhang, H. L. Yang, and Z. Yang, "An adaptive adversarial domain adaptation approach for corn yield prediction," *Comput. Electron. Agricult.*, vol. 187, Aug. 2021, Art. no. 106314.

[75] F. Magistri, J. Weyler, D. Gogoll, P. Lottes, J. Behley, N. Petrinic, and C. Stachniss, "From one field to another—Unsupervised domain adaptation for semantic segmentation in agricultural robotics," *Comput. Electron. Agricult.*, vol. 212, Sep. 2023, Art. no. 108114.

[76] Y. Hu, G. Cheng, W. Chen, and B. Jiang, "Attribute-based zero-shot learning for encrypted traffic classification," *IEEE Trans. Netw. Service Manage.*, vol. 19, no. 4, pp. 4583–4599, Dec. 2022.

[77] E. Kodirov, T. Xiang, Z. Fu, and S. Gong, "Unsupervised domain adaptation for zero-shot learning," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2452–2460.

[78] Q. Wang, P. Bu, and T. P. Breckon, "Unifying unsupervised domain adaptation and zero-shot visual recognition," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8.

[79] W. Y. Jhoo and J.-P. Heo, "Collaborative learning with disentangled features for zero-shot domain adaptation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 8876–8885.

[80] X. Li, M. Fang, and B. Chen, "Generalized zero-shot domain adaptation with target unseen class prototype learning," *Neural Comput. Appl.*, vol. 34, no. 20, pp. 17793–17807, Oct. 2022.

[81] S. Sankaranarayanan, Y. Balaji, C. D. Castillo, and R. Chellappa, "Generate to adapt: Aligning domains using generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8503–8512.

[82] J. Feng, S. Jegelka, S. Yan, and T. Darrell, "Learning scalable discriminative dictionary with sample relatedness," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1645–1652.

[83] C.-A. Hou, Y. H. Tsai, Y.-R. Yeh, and Y. F. Wang, "Unsupervised domain adaptation with label and structural consistency," *IEEE Trans. Image Process.*, vol. 25, no. 12, pp. 5552–5562, Dec. 2016.

[84] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata, "Zero-shot learning—A comprehensive evaluation of the good, the bad and the ugly," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 9, pp. 2251–2265, Sep. 2019.

[85] Y. Luo, P. Liu, T. Guan, J. Yu, and Y. Yang, "Adversarial style mining for one-shot unsupervised domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 20612–20623.

[86] R. K. Sanodiya, J. Mathew, B. Paul, and B. A. Jose, "A kernelized unified framework for domain adaptation," *IEEE Access*, vol. 7, pp. 181381–181395, 2019.

[87] S. K. Zhou and R. Chellappa, "From sample similarity to ensemble similarity: Probabilistic distance measures in reproducing kernel Hilbert space," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 6, pp. 917–929, Jun. 2006.

[88] Y. Gulzar, "Fruit image classification model based on MobileNetV2 with deep transfer learning technique," *Sustainability*, vol. 15, no. 3, p. 1906, Jan. 2023.

[89] F. Zhang, J. Bai, J. Zhang, Z. Xiao, and C. Pei, "An optimized training method for GAN-based hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 10, pp. 1791–1795, Oct. 2021.

[90] Y. H. Tsai, Y.-R. Yeh, and Y. F. Wang, "Learning cross-domain landmarks for heterogeneous domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5081–5090.

[91] J. Zhang, W. Li, and P. Ogunbona, "Joint geometrical and statistical alignment for visual domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5150–5158.

[92] A. Lengyel, S. Garg, M. Milford, and J. C. van Gemert, "Zero-shot day-night domain adaptation with a physics prior," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 4379–4389.

[93] I. Goodfellow, "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[94] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*.

[95] C. Chen, Z. Chen, B. Jiang, and X. Jin, "Joint domain alignment and discriminative feature learning for unsupervised deep domain adaptation," in *Proc. Nat. Conf. Artif. Intell.*, vol. 33, no. 1, 2019, pp. 3296–3303.

[96] G. Wen, H. Chen, D. Cai, and X. He, "Improving face recognition with domain adaptation," *Neurocomputing*, vol. 287, pp. 45–51, Apr. 2018.

[97] H. Rehana, M. Ibrahim, and M. H. Ali, "Plant disease detection using region-based convolutional neural network," 2023, *arXiv:2303.09063*.

[98] A. S. Paymode and V. B. Malode, "Transfer learning for multi-crop leaf disease image classification using convolutional neural network VGG," *Artif. Intell. Agricult.*, vol. 6, pp. 23–33, Jan. 2022.

[99] M. Kaur and R. Bhatia, "Development of an improved tomato leaf disease detection and classification method," in *Proc. IEEE Conf. Inf. Commun. Technol.*, Dec. 2019, pp. 1–5.

[100] V. Tiwari, R. C. Joshi, and M. K. Dutta, "Dense convolutional neural networks based multiclass plant disease detection and classification using leaf images," *Ecolog. Informat.*, vol. 63, Jul. 2021, Art. no. 101289.

[101] R. Gandhi, S. Nimbalkar, N. Yelamanchili, and S. Ponkshe, "Plant disease detection using CNNs and GANs as an augmentative approach," in *Proc. IEEE Int. Conf. Innov. Res. Develop. (ICIRD)*, May 2018, pp. 1–5.

[102] D. P. Rubanga, "A deep learning approach for determining effects of Tuta Absoluta in tomato plants," 2020, *arXiv:2004.04023*.

[103] R. Katafuchi and T. Tokunaga, "LEA-Net: Layer-wise External Attention Network for Efficient Color Anomaly Detection," 2021, *arXiv:2109.05493*.

[104] Y. Lu, D. Chen, E. Olaniyi, and Y. Huang, "Generative adversarial networks (GANs) for image augmentation in agriculture: A systematic review," *Comput. Electron. Agricult.*, vol. 200, Sep. 2022, Art. no. 107208.

[105] J. G. M. Esgario, R. A. Krohling, and J. A. Ventura, "Deep learning for classification and severity estimation of coffee leaf biotic stress," *Comput. Electron. Agricult.*, vol. 169, Feb. 2020, Art. no. 105162.

[106] J. P. S. Schuler, S. Romani, M. Abdel-Nasser, H. Rashwan, and D. Puig, "Color-aware two-branch DCNN for efficient plant disease classification," *Mendel*, vol. 28, no. 1, pp. 55–62, Jun. 2022.

[107] M. Brahimi, S. Mahmoudi, K. Boukhalfa, and A. Moussaoui, "Deep interpretable architecture for plant diseases classification," in *Proc. Signal Process., Algorithms, Archit., Arrangements, Appl. (SPA)*, Sep. 2019, pp. 111–116.

[108] H. Nazki, S. Yoon, A. Fuentes, and D. S. Park, "Unsupervised image translation using adversarial networks for improved plant disease recognition," *Comput. Electron. Agricult.*, vol. 168, Jan. 2020, Art. no. 105117.

[109] E. Bollis, H. Pedrini, and S. Avila, "Weakly supervised learning guided by activation mapping applied to a novel citrus pest benchmark," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 310–319.

[110] A. Prasad, N. Mehta, M. Horak, and W. D. Bae, "A two-step machine learning approach for crop disease detection using GAN and UAV technology," *Remote Sens.*, vol. 14, no. 19, p. 4765, Sep. 2022.

[111] S. Jha, "Crop disease classification using support vector machines with green chromatic coordinate (GCC) and attention based feature extraction for IoT based smart agricultural applications," 2023, *arXiv:2311.00429*.

R. SATYA RAJENDRA SINGH (Member, IEEE) is currently pursuing the Ph.D. degree in computer science and engineering. He is a Research Scholar with the Indian Institute of Information Technology (IIIT) Sri City. He published and contributed to reputable journals and conferences. His research interests include computer vision, machine learning, pattern recognition, transfer learning, and domain adaptation.

RAKESH KUMAR SANODIYA (Member, IEEE) received the Ph.D. degree from the Indian Institute of Technology (IIT), Patna, in 2019, where he developed state-of-the-art domain adaptation approaches for image classification. He was a Research Assistant Professor with the National Taipei University of Technology, Taiwan. He is currently an Assistant Professor with the Indian Institute of Information Technology Sri City, Chittoor. He is an accomplished academician with expertise in the field of domain adaptation for image classification. His research has focused on developing feature and instance re-weighting based on semi-supervised and unsupervised domain adaptation frameworks for visual image classification. He has published his research results in several reputable journals, including IEEE TRANSACTIONS ON ARTIFICIAL INTELLIGENCE, *Knowledge-Based Systems*, *Expert Systems With Applications*, *Soft Computing*, *Applied Intelligence*, *Image and Vision Computing*, *Sensors*, and IEEE ACCESS, among others. He has presented his research findings at various prestigious conferences, such as IEEE CEC, IEEE IJCNN, ICONIP, INDICON, and many others.

• • •