Convolutional Ensembling based Few-Shot Defect Detection Technique

Soumyajit Karmakar
IIIT Guwahati
Guwahati, Assam, India

Abeer Banerjee
CSIR-CEERI
Pilani, Rajasthan, India

Sanjay Singh
CSIR-CEERI
Pilani, Rajasthan, India

ABSTRACT

Over the past few years, there has been a significant improvement in the domain of few-shot learning. This learning paradigm has shown promising results for the challenging problem of anomaly detection, where the general task is to deal with heavy class imbalance. Our paper presents a new approach to few-shot classification, where we employ the knowledge-base of multiple pre-trained convolutional models that act as the backbone for our proposed few-shot framework. Our framework uses a novel ensembling technique for boosting the accuracy while drastically decreasing the total parameter count, thus paving the way for real-time implementation. We perform an extensive hyperparameter search using a power-line defect detection dataset and obtain an accuracy of 92.30% for the 5-way 5-shot task. Without further tuning, we evaluate our model on competing standards with the existing state-of-the-art methods and outperform them.

KEYWORDS

Few-Shot Classification, Anomaly Detection, Ensemling Strategy

1 INTRODUCTION

In conventional deep-learning-based computer vision approaches, one can observe a positive relationship between the size of the training dataset and the performance of the model. In contrast, the few-shot based approaches attempt to achieve similar performances while using significantly lesser training dataset.

There has been a lot of recent research in this domain [7–9, 16–18, 22, 35]. The major benefit of a few-shot learning based solution to a computer vision problem, say image classification, is that the overall computation cost of achieving a certain level of accuracy is drastically lower as compared to traditional data-driven approaches. As few-shot based approach requires a few examples of a per class by definition, the task of data collection and annotation becomes significantly easier. This attribute makes it perfectly suitable for dealing with problems where the collection of data for a particular class is either difficult or the event in consideration is naturally rare. Researchers have been employing few-shot learning in the problem of anomaly detection [5, 19, 27] as anomalies are naturally rare, which creates a huge class imbalance. While there have been approaches to solve the problem of class imbalance with synthetic data generation using generative adversarial networks [25, 34], they still suffer from drawbacks such as, huge computational cost involved during training.

One of the most popular frameworks for this task is meta-learning. Here the model focuses on learning to learn, rather than memorizing the particular features of images. This helps in enabling the model to distinguish objects without the requirement of a huge dataset. A few-shot problem is usually defined by the pair n-way

k-shot, where n refers to the number of classes in question and k refers to the number of examples in each class on which the model is trained. The training set thus formed is called the support set, and the testing set is called the query set.

For training the few-shot learner, two commonly used approaches are gradient-based approach and metric-based approach. In the case of gradient-based approaches, the base model is updated as a trainable function [1] and the gradients are then back-propagated across it [8, 20]. In metric-based approaches, a feature embedding is learnt which is then used to classify the query images based on a similarity function [10, 29].

In this paper, we present a new approach, influenced by the work of Chowdhury et al. [3]. Chowdhury et al. employed a huge combination of library-learners which are basically pre-trained CNNs available off-the-shelf, and used them to parallelly compute the feature embeddings. They combined the features using simple ensembling techniques and passed it to a multi-layered perceptron. Their approach simplifies the training process but involves a huge number of parameters to achieve an acceptable amount of accuracy. We develop a new ensembling strategy that involves the use of a convolutional block to stack and combine the features obtained from each feature extractor. This drastically reduces the parameter counts and boosts the classification accuracy. We evaluate the reliability of our method on a powerline components dataset with huge class imbalance. This dataset has been collected by drones and has five major classes namely, insulator, nest, bolt, spacer and the anomaly class, missing bolt. The dataset contains thousands of images of multiple resolutions in all classes except in the anomaly class which contains significantly less number of examples. This dataset emulates the practical scenarios perfectly as the captured images are mostly of low resolutions which makes the classification task particularly challenging.

2 STRATEGY

We took some well-known off-the-shelf convolutional neural networks, ResNet [11], DenseNet [13], Inception [30], Xception [2], EfficientNet [31], all trained on ILSVRC2012 [26] and discarded the fully-connected layers to obtain their respective convolutional segments. These convolution subnetworks are used to extract and form the feature embeddings corresponding to each image. We reshaped the obtained features to a stack of channels and passed it to our proposed model for the few-shot classification task. We experimented with multiple pre-trained CNNs and found that a combination of three such networks had provided the best results. A detailed performance evaluation report using various off-the-shelf pre-trained models has been provided in Section 3.

The support and query sets were generated by randomly sampling our dataset. The few-shot training was performed using the
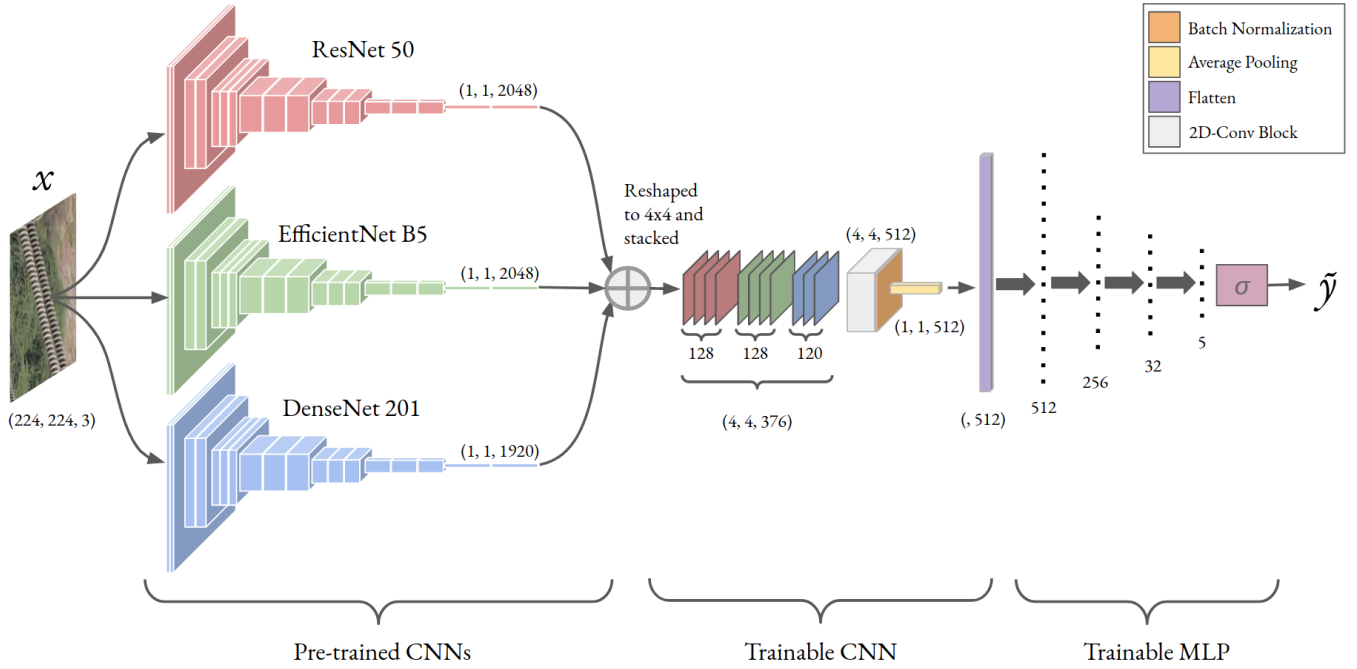
Figure 1: Architecture

support set that comprised of a few examples for each class while the few-shot query set was used to evaluate the model performance. The model architecture and the training details are discussed in the subsequent sections.

2.1 Architecture

For our best performing approach, we have used the combination of ResNet 50, EfficientNet B5, and DenseNet 201 for computing the feature embeddings corresponding to each image. The obtained features were reshaped to 4×4 spatial blocks and stacked to form the input for our proposed model. The stacked channels form the input x, that is passed through a convolutional block comprising of $[Conv2D \longrightarrow ReLU \longrightarrow BatchNorm \longrightarrow AvgPool]$. The output of this block is then flattened and passed through a Multi-Layer Perceptron with two hidden layers with 256 neurons and 32 neurons. The output layer has five neurons for providing a 5 class classification which is then passed through a softmax layer to obtain the final classified label $\tilde{y}$. The detailed architecture is illustrated in Figure 1 and the model summary is provided in Table 1.

2.2 Training and Implementation

This dataset consists of five classes, i.e., $n = 5$. We experimented with different numbers of examples per class (k) that can be used for training the few shot learner and found that the model has an optimum performance at k=5. We randomly selected 32 images from each class from the whole dataset and split them into two groups of 5 and 27 images for the support and query sets, respectively. The support set for the few-shot training process was created by combining the extracted features from the pre-trained networks with their associated labels. The same procedure was followed for

Layers	Output Shape	Parameters
ResNet 50	$1 \times 1 \times 2048$	23.6M (Frozen)
EfficientNet B5	$1 \times 1 \times 2048$	28.5M (Frozen)
DenseNet 201	$1 \times 1 \times 1920$	18.3M (Frozen)
Concat	$1 \times 1 \times 6016$	-
Reshape stack	$4 \times 4 \times 376$	-
Conv2D	$4 \times 4 \times 512$	1.7M
BatchNorm	$4 \times 4 \times 512$	2k
AvgPool	$1 \times 1 \times 512$	-
Flatten	512	-
Dense In	512	262k
Hidden Dense 1	256	131k
Hidden Dense 2	32	8k
Dense Out	5	0.1k
Total Trainable Parameters		2.1M

Table 1: Architecture Table

the rest of the 27 images, except that their corresponding labels were not supplied, and then the resulting features formed the query set.

Figure 2 illustrates the training pipeline using three pre-trained networks as the feature extractors. We use the proposed feature ensembling strategy to combine the extracted features which are passed to the trainable CNN layers and further propagated to the trainable MLP layer to obtain the classified output. The reshaping and stacking technique is explained visually in the same figure. The network minimizes the categorical cross-entropy loss L_{CCE} which is backpropagated through the trainable layers. We use Adam
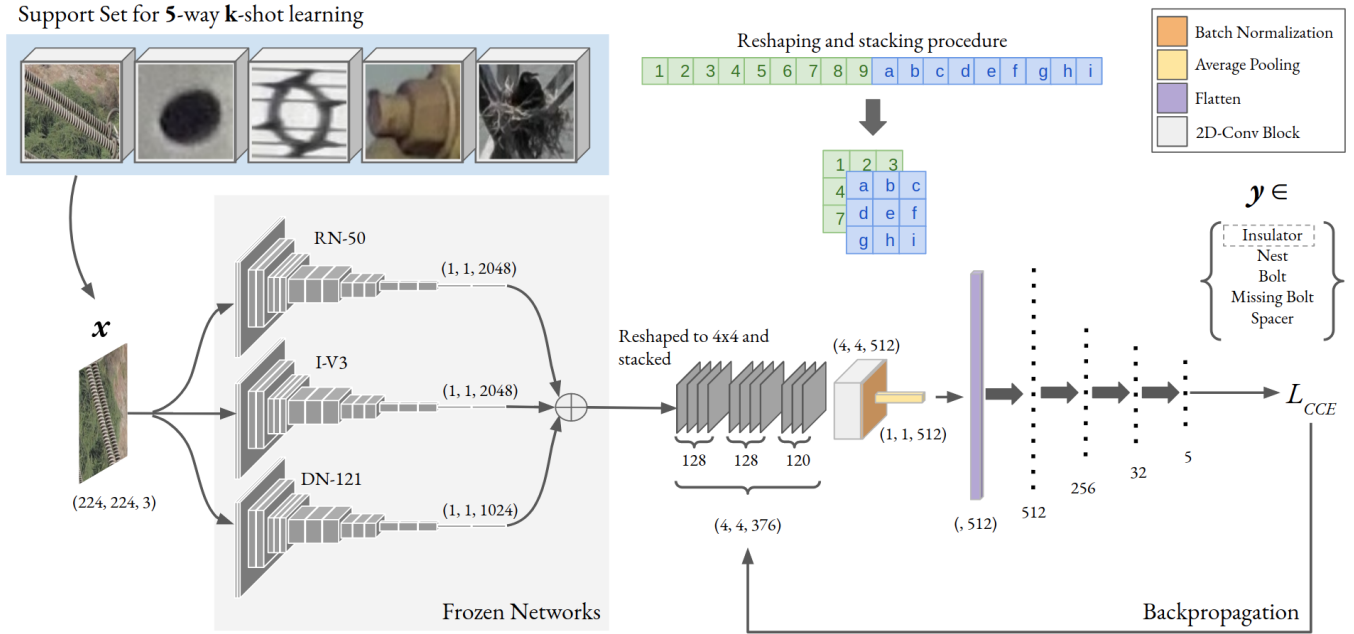
Figure 2: Training Pipeline

optimizer [14] with a learning rate of 5×10^{-5}. We use values for the L_2 regularization constant as high as 0.5 to ensure that the model does not overfit to the training data. The network takes 300 epochs for the loss value to saturate, but as the execution time of each epoch is less than a few milliseconds, the overall process does not take more than a few seconds to complete. All training and testing were performed on a system powered by an Intel Xeon 2.90 GHz quad-core CPU with NVIDIA 1080 GPU having 8GB of graphics memory.

3 ABLATION STUDY AND HYPERPARAMETER SEARCH

We performed an extensive ablation study to ensure the reliability of our approach. We ablate our model in terms of the feature extractors, the ensembling strategy and the hyperparameters for the trainable CNN and MLP network. We considered eight different backbone networks, each with less than 30M parameters, and 4 different kernel sizes for the ensembling. For studying the ablation of ensembling techniques, we fix the structure of the trainable CNN-MLP model on a trial-error basis to observe the general trend in accuracy. We later on refine this structure based on further ablation studies. For all the testing purposes we have used k-fold cross verification to obtain a reliable performance score.

3.1 Ablation study of ensembling techniques

We begin with using only one pre-trained network as the backbone. The performance of each of the models corresponding to various kernel sizes for ensembling is presented in Table 2. A clear trend can be observed that the 4×4 kernel size works the best for each model. We pick three of the best performing models for further ablation studies. Since the output shape of most of the networks

Backbone	32×32	16×16	8×8	4×4
ResNet 50	87.86	86.57	85.53	88.02
ResNet 50 V2	82.32	85.74	86.28	87.98
DenseNet 121	76.39	78.47	77.78	84.22
DenseNet 201	-	-	79.04	87.53
Inception V3	64.67	63.87	69.02	75.37
Xception	75.48	71.73	73.93	78.27
EfficientNet V2S	-	78.83	78.56	82.79
EfficientNet B5	77.56	79.77	75.91	82.76

Table 2: Ablation Study of Reshaped kernel size for ensembling.

is either 1024 or 2048, we can easily convert it into 1 or 2 stacks of 32×32. The output size of DenseNet 201 is 1920 which can only be converted into stacks of 8×8 or smaller, therefore, some of the columns in Table 2 are missing. For similar reasons, some accuracies corresponding to EfficientNet V2S is also missing.

3.2 Ablation Study of CNN-MLP network

Here we study the effects of changing the structure of the trainable CNN-MLP model. We fix the backbone network as the best performing model and the kernel size to 4×4 as obtained in Table 2. We experimented with multiple numbers of hidden layers in the MLP each with varying number of neurons, including the case with no hidden layer at all. The results are listed in Table 3. The best result was obtained for the case when there were 2 hidden layers, with 256 and 32 neurons respectively. The input layer of the MLP depends on the output of the CNN block. From this we conclude the optimal number of filters in the CNN to be 512.
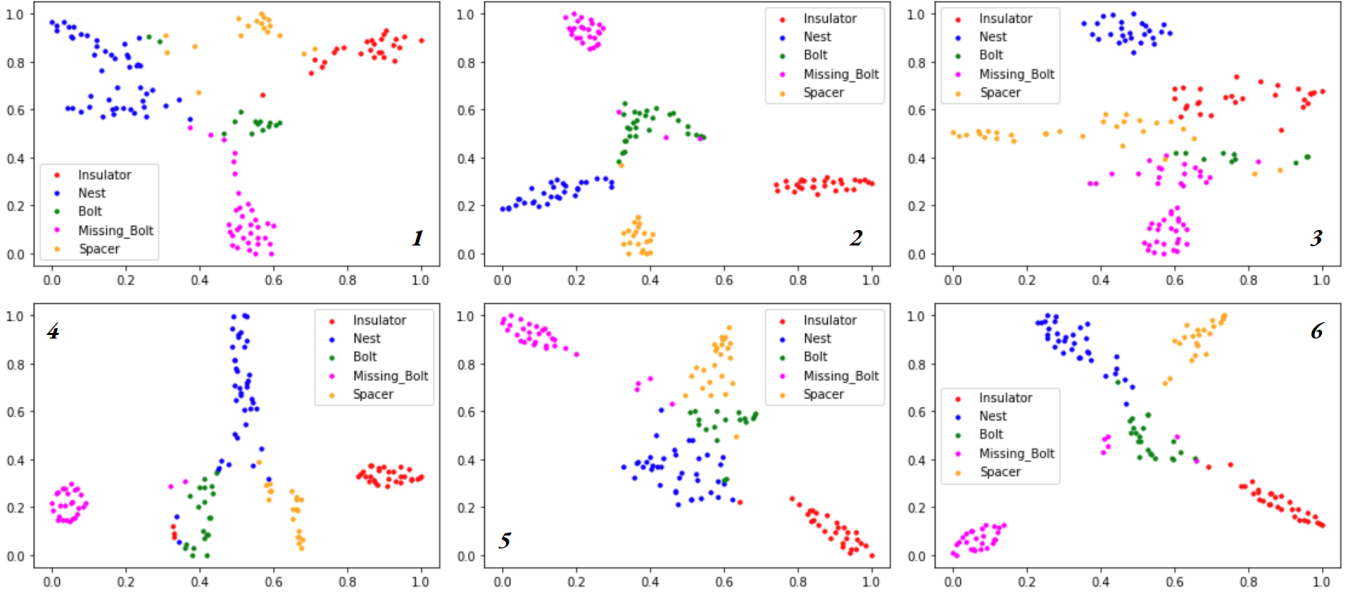
3

Figure 3: Ablation study of the proposed framework using TSNE plots: (1) DenseNet 201 only. (2) ResNet 50 only. (3) EfficientNet B5 only. (4) ResNet 50 + DenseNet 201. (5) DenseNet 201 + EfficientNet B5. (6) ResNet 50 + EfficientNet B5.
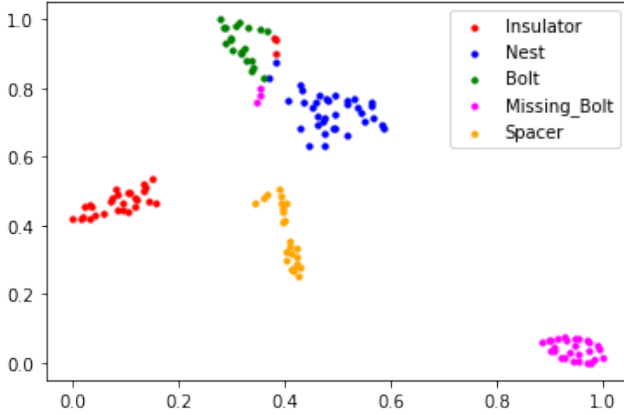

Figure 4: Ablation study of the proposed framework using TSNE plots: DenseNet 201 + ResNet 50 + EfficientNet B5 combined.

3.3 Ablation study of number of feature extractors

In this section we study the impact of the combining multiple models together. We select the three best performing models from the previous sections[1]. We compare the performances of the models taken one at a time, two at a time, and all three at a time. Considering the practical memory constraints of most mobile devices, we limited our study to a maximum of three models taken together to restrict the total parameter count. Table 4 lists the accuracies thus obtained.

---

[1]We experimented with multiple combinations involving the other models as well, but the combination of the three best models produced the best results.

Hidden Layers	Structure	Accuracy (%)
0	$1024 \rightarrow 5$	57.56
	$512 \rightarrow 5$	54.09
1	$1024 \rightarrow 128 \rightarrow 5$	65.33
	$512 \rightarrow 128 \rightarrow 5$	70.68
	$256 \rightarrow 128 \rightarrow 5$	66.68
2	$1024 \rightarrow 512 \rightarrow 128 \rightarrow 5$	82.93
	$512 \rightarrow 256 \rightarrow 32 \rightarrow 5$	89.08
	$256 \rightarrow 128 \rightarrow 16 \rightarrow 5$	73.10
	$128 \rightarrow 64 \rightarrow 16 \rightarrow 5$	73.25

Table 3: Ablation Study of CNN-MLP structure with ResNet 50 as the backbone and kernel size = 4×4.

Backbone	Accuracy (%)
RN50	88.24
DN201	87.53
ENB5	82.76
RN50 + DN201	89.39
ENB5 + DN201	90.12
RN50 + ENB5	90.95
RN50 + DN201 + ENB5	92.30

Table 4: Ablation Study of Feature Extractors for ensembling strategy with a kernel size = 4×4.

We can observe a clear performance improvement as we increase the number of pre-trained networks for feature extraction.

The class separability of the combinations is visualised using a TSNE [32] plot given in Figure 3. A t-distributed stochastic neighbour embedding, or TSNE, is a dimensionality reduction tool that
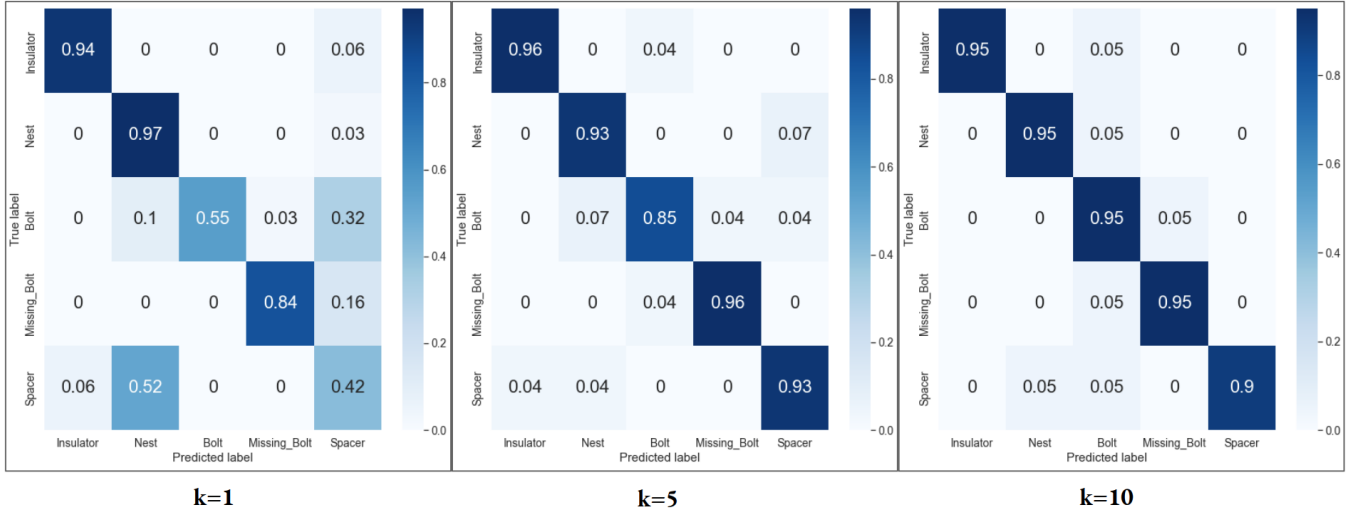
Figure 5: Confusion matrix obtained using different values of k in 5-way k-shot.

helps in visualizing the clustering ability of a model. We observe that when we take the backbones one at a time (1, 2, and 3), the models fail to form sharp clusters, thus having the lowest accuracy. The clustering capability of the model improves as we increase the number of backbone networks. Figure 4 shows the sharp clustering capability of the proposes model.

4 COMPARISON WITH THE STATE-OF-THE-ART

The final results of our model is presented in this section. We provide the performance scores on the Powerline components dataset first and then compare our model with other popular models on some standard datasets.

4.1 Results on Powerline Dataset

All the testing was done by running the model multiple times and using k-fold cross validation method to get an average score. Figure 6 shows some examples of images that were correctly classified by the model. It is to be noted that the images were of different resolutions. They are rescaled to the same size for display purposes.

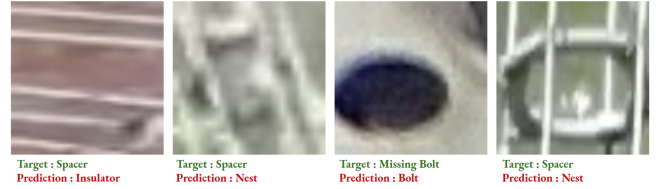**Figure 6: Some examples of correctly classified images.**

Figure 7: Some examples of incorrectly classified images.

k	1	5	10
Accuracy (%)	70.11	92.30	95.42

Table 5: Results obtained using our best model for different values of k in 5-way k-shot.

Figure 7 shows some of the misclassified images. It can be observed that most of the misclassifications were due to heavy amount of noise and blur.

Table 5 lists the results obtained by our best model with the three feature extractors namely, ResNet 50, DenseNet 201, EfficientNet B5. The ensembling strategy used a kernel size of 4×4 and 512 filters for the CNN block, and two hidden layers ($256 \rightarrow 32$) in the MLP block. The results were obtained by varying the number of training examples in each class. Figure 5 contains the confusion matrices for the three values of k. For $k = 1$, the model was supplied with only one training image per class, explaining the sharp drop in accuracy.

4.2 Results on Standard Datasets

We compare our model with the existing state-of-the-art methods on various datasets, such as the Aircraft [21], Traffic [24], Omniglot [15], FC100 [12], VGG Flower [23], and the Texture [4]. As we perform our hyperparameter search on the power-line anomaly dataset containing five classes only, we stick to the results for the 5-way

Model	Aircraft	Traffic	Omniglot	Texture	FC100	VGG Flower
MAML	33.1 ± 0.6	67.4 ± 0.9	82.6 ± 0.7	56.9 ± 0.8	62.0 ± 0.8	78.0 ± 0.7
MatchingNet	33.5 ± 0.6	73.7 ± 0.8	89.7 ± 0.5	54.7 ± 0.7	59.4 ± 0.8	74.2 ± 0.8
ProtoNet	41.5 ± 0.7	75.0 ± 0.8	95.5 ± 0.3	62.9 ± 0.7	64.7 ± 0.8	86.7 ± 0.6
SUR	45.2 ± 0.8	70.6 ± 0.8	98.7 ± 0.1	59.6 ± 0.7	67.2 ± 1.0	90.8 ± 0.5
Chowdhury et al.	**68.9 ± 0.9**	85.8 ± 0.7	98.0 ± 0.2	85.7 ± 0.6	80.5 ± 0.6	97.9 ± 0.2
Ours	65.6 ± 1.7	**93.1 ± 0.3**	**99.0 ± 0.3**	**86.8 ± 0.6**	**91.4 ± 0.2**	**98.8 ± 0.3**

Table 6: Comparative analysis of our model with the state-of-the-art methods for the 5-way 5-shot problem.

5-shot problem. Table 6 shows a detailed comparative study of our method with the existing state-of-the-art methods. For comparison we chose some of the most popular existing alternative techniques for few-shot classification, such as, MAML [8], MatchingNet [33], ProtoNet [28], SUR [6] and the model proposed by Chowdhury et. al. [3]. It can be observed that under most circumstances, our method is able to outperform the model by Chowdhury et al., our inspiration, by a significant margin for most datasets.

5 CONCLUSION

In this paper we experimented a new approach for few-shot image classification. We evaluated our approach on a powerline anomaly dataset where the anomaly class was "missing bolts". We developed an ensembling technique that combines the extracted features of different pre-trained networks in a parameter efficient way. The classification accuracy obtained by training the model with a 5-way k-shot support set was above 90% for $k \geq 5$. After extensive performance evaluation with multiple combinations of feature extractors, we found that the accuracy score was obtained with a strategic combination of three specialized pre-trained networks. We visualized the class separability of our method using TSNE plots and confusion matrices and finally obtained a peak classification accuracy of 92.30% for 5-way k-shot task. The dataset used to evaluate our framework was new and challenging because it included realistic images of multiple resolutions. A major critique of our approach is the sensitivity of the accuracy on each image of the support set. For example, as the support set was randomly selected, there were cases where the all the samples under a particular class were similar to each other and failed to represent other variations thereby compromising the overall accuracy. Therefore, the selection of support set should be done with extreme care. The complete code will be made publicly available for further research.

ACKNOWLEDGMENTS

REFERENCES

[1] Yoshua Bengio, Renato De Mori, Giovanni Flammia, and Ralf Kompe. 1992. Global optimization of a neural network-hidden Markov model hybrid. *IEEE transactions on Neural Networks* 3, 2 (1992), 252–259.
[2] François Chollet. 2017. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1251–1258.
[3] Arkabandhu Chowdhury, Mingchao Jiang, Swarat Chaudhuri, and Chris Jermaine. 2021. Few-shot image classification: Just use a library of pre-trained feature extractors and a simple classifier. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9445–9454.
[4] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. 2014. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3606–3613.
[5] Kaize Ding, Qinghai Zhou, Hanghang Tong, and Huan Liu. 2021. Few-shot network anomaly detection via cross-network meta-learning. In *Proceedings of the Web Conference 2021*. 2448–2456.
[6] Nikita Dvornik, Cordelia Schmid, and Julien Mairal. 2020. Selecting relevant features from a multi-domain representation for few-shot classification. In *European Conference on Computer Vision*. Springer, 769–786.
[7] Li Fei-Fei, Robert Fergus, and Pietro Perona. 2006. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence* 28, 4 (2006), 594–611.
[8] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*. PMLR, 1126–1135.
[9] Chelsea Finn, Kelvin Xu, and Sergey Levine. 2018. Probabilistic model-agnostic meta-learning. *Advances in neural information processing systems* 31 (2018).
[10] Spyros Gidaris and Nikos Komodakis. 2018. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4367–4375.
[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.
[12] Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel. 2013. Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark. In *The 2013 international joint conference on neural networks (IJCNN)*. Ieee, 1–8.
[13] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4700–4708.
[14] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[15] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. 2015. Human-level concept learning through probabilistic program induction. *Science* 350, 6266 (2015), 1332–1338.
[16] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. 2019. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10657–10665.
[17] Yoonho Lee and Seungjin Choi. 2018. Gradient-based meta-learning with learned layerwise metric and subspace. In *International Conference on Machine Learning*. PMLR, 2927–2936.
[18] Hongyang Li, David Eigen, Samuel Dodge, Matthew Zeiler, and Xiaogang Wang. 2019. Finding task-relevant features for few-shot learning by category traversal. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 1–10.
[19] Yiwei Lu, Frank Yu, Mahesh Kumar Krishna Reddy, and Yang Wang. 2020. Few-shot scene-adaptive anomaly detection. In *European Conference on Computer Vision*. Springer, 125–141.
[20] Dougal Maclaurin, David Duvenaud, and Ryan Adams. 2015. Gradient-based hyperparameter optimization through reversible learning. In *International conference on machine learning*. PMLR, 2113–2122.
[21] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. 2013. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151* (2013).
[22] Alex Nichol and John Schulman. 2018. Reptile: a scalable metalearning algorithm. *arXiv preprint arXiv:1803.02999* 2, 3 (2018), 4.
[23] Maria-Elena Nilsback and Andrew Zisserman. 2008. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*. IEEE, 722–729.

[24] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. 2018. Tadam: Task dependent adaptive metric for improved few-shot learning. *Advances in neural information processing systems* 31 (2018).

[25] Ahmad B Qasim, Ivan Ezhov, Suprosanna Shit, Oliver Schoppe, Johannes C Paetzold, Anjany Sekuboyina, Florian Kofler, Jana Lipkova, Hongwei Li, and Bjoern Menze. 2020. Red-GAN: Attacking class imbalance via conditioned generation. Yet another medical imaging perspective.. In *Medical Imaging with Deep Learning.* PMLR, 655–668.

[26] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision* 115, 3 (2015), 211–252.

[27] Shelly Sheynin, Sagie Benaim, and Lior Wolf. 2021. A hierarchical transformation-discriminating generative model for few shot anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 8495–8504.

[28] Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. *Advances in neural information processing systems* 30 (2017).

[29] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. 2018. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 1199–1208.

[30] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 1–9.

[31] Mingxing Tan and Quoc Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning.* PMLR, 6105–6114.

[32] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[33] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. *Advances in neural information processing systems* 29 (2016).

[34] Hao Yang and Yun Zhou. 2021. IDA-GAN: a novel imbalanced data augmentation GAN. In *2020 25th International Conference on Pattern Recognition (ICPR).* IEEE, 8299–8305.

[35] Jaesik Yoon, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. 2018. Bayesian model-agnostic meta-learning. *Advances in neural information processing systems* 31 (2018).