FUSION OF DEEP TRANSFER LEARNING WITH MIXED CONVOLUTION NETWORK

Anonymous authors
Paper under double-blind review

ABSTRACT

Global thirst in computer vision for image classification is performance improvement and parameter optimization. The evolution of deep learning raised the boundaries of the model size to hundreds of millions of parameters. This obliquely influences the training time of the model. Thus, contemporary research has diverted to parameter optimization par with performance. In this paper, a fusion-based deep transfer learning approach has been furbished with a mixed convolution block. The proposed mixed convolution block has been designed using two convolution paths including residual and separable convolutions. The residual convolution avoids vanishing gradient while separable convolution includes depthwise features. The experiments on the popular Fashion-MNIST bench mark dataset have proved that the proposed mixed convolution enticed the pre-trained models. It has been observed that there is a clear improvement of 1% than the base models. Further, the proposed fusion model exhibits a competing performance of 96.04% with existing models.

1 INTRODUCTION

Recently, deep convolutional neural networks became popular due to their magnanimous performance as compared to traditional machine learning. However, computer vision classification research still acquiring new approaches for performance improvement. Several factors influence the performance of the model which can be segregated as follows.

- **Design parameters:** It includes the depth and width of the network and various other design parameters including the concept of design blocks, attention mechanisms, etc.
- **Training parameters:** It includes the optimizer, loss function, data augmentation, and other parameters.

Initial stages of deep learning, researchers have started to design new networks for better performance. ResNet (He et al., 2016), InceptionNet (Szegedy et al., 2015), InceptionResNet (Szegedy et al., 2016), DenseNet (Huang et al., 2017), and EfficientNet (Tan & Le, 2019) are widely used deep networks in this era. In addition to the pre-trained models, the researchers have focused on training parameters for polishing classification results. A few of the contemporary proposals are as follows. Differential Architecture Search (DARTS) has been performed by Liu et al. (2018) to find a computationally light method. Later, Tanveer et al. (2021) have fine-tuned the DARTS approach. Phong & Ribeiro (2020) have integrated RNN as additional layers for model recognition. They also have developed multimodel end-to-end ensembling for expert predictions. Jeevan et al. (2022) have mixed spatial information from pixels using a 2D-discrete wavelet transform. Sun et al. (2020) have extracted neural network-based models from dynamic data using ordinary and partial differential equations. Rajasegaran et al. (2019) have designed a 3D convolution-based dynamic routing algorithm for a deep capsule network. Zinchenko & Lishudi (2022) have proposed an ensemble algorithm for neural networks using a 3D convolution-based dynamic routing algorithm.

Foret et al. (2020) have experimented with the geometry of the loss landscape and generalization. They have proposed Sharpness-Aware Minimization (SAM) using neighborhoods parameter search. Another training optimization approach has been proposed by Nøkland & Eidnes (2019). They have generated local error signals using two different supervised loss functions. Zhong et al. (2017) have introduced a new random erasing approach for data augmentation. A rectangle region in an

image has been randomly selected and is erased with random values. Similarly, Harris et al. (2020) have demonstrated various Mixed Sample Data Augmentation (MSDA) including MixUp, CutMix, and FMix. In a broad view, the models have enlightened the training approach with improved optimizers and augmentation techniques. A few of them have focused on re-design of some network components for better adaptation of features. This work is also motivated to utilize the fine-tuning of existing models through a feature enhancement network. The performance of the resulting model has improved due to domain specific feature learning through feature enhancement network.

2 METHODOLOGY

In general, the existing deep convolutional neural network models are treated as a single network with two sub-networks. Firstly, the local features of the images are extracted with a deep convolution network. Then, the final classification results are produced with a fully connected neural network as shown in figure 1 (a). This paper introduces a new approach to enhancing the power of pre-trained models with an ensemble approach. The existing pre-trained models have trained with the Imagenet dataset for thousand-class classification. These weights can be re-used as it is through transfer learning if the problem is same. The transfer learning is a proven approach for many classification applications with improved performance. However, transfer learning has admired several researchers to invoke the pre-trained models for their custom classification problems. In such cases, the pre-trained models are re-trained with the custom dataset with customized output layers. With this, we can conclude that if there are any changes in the problem, then the network needs re-designing and/or re-training. A huge number of proposals have utilized this approach for classification over a wide range of applications. A few recent of them are (Liu et al., 2020), (Wang et al., 2019), and (Shamsi et al., 2021). However, the pre-trained models can be further improved by embedding another network known as a feature enhancement network as shown in figure 1 (b). The motivation and core idea of the feature enhancement network along with details of proposed ensemble and fusion models are as follows.

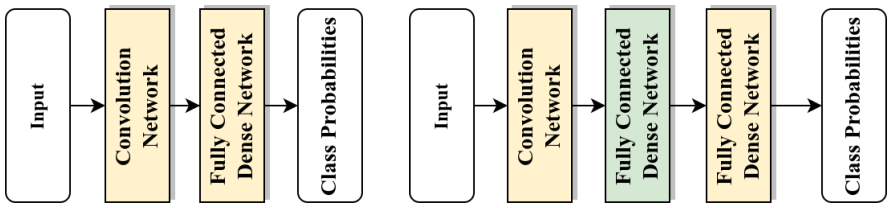

Figure 1: View of CNN model; (a) General view of CNN model ; (b) Proposed view of CNN model.

2.1 FEATURE ENHANCEMENT NETWORK

The existing networks focus on feature learning and classification networks. On the other hand, the proposed model insights feature enhancement through a short custom network. The pre-trained networks imbibe generalized features with ImageNet and are weak to perform domain-specific tasks. It needs a short custom network to capture the domain-specific features. Thus, a mixed convolution network with two parallel convolution blocks has designed to address the problem. The details of each block are as follows.

- **Residual Convolution Block (RCB)**:
  This block utilizes residual connection with convolution layers, and hence it is named a residual convolution block. It consists of three convolution layers having a residual connection from the first layer to the third layer. These convolution layers use ReLU activation with diminished filters as 160. Then another convolution with leaky ReLU activation has performed to regain its filter count to 1280. It acts as a filter auto-encoder to adapt the domain-specific feature learning. This block mainly focuses on adhering filter-oriented features to the network with a resolution of (3, 3) as shown in figure 2 (a).

- **Separable Convolution Block (SCB)**:
  This block involves separable convolution with a pooling mechanism. It consists of two separable convolution layers included with average pooling. The main intuition to utilize separable convolution is to involve point-wise convolution. The point-wise convolution iterates through every single point of the feature map. It also utilizes leaky ReLU activated separable layer to regain the original filter count like a residual convolution block as shown in figure 2 (b).

Both of the blocks include a global average pooling to obtain a one-dimensional feature vector as output. The global average pooling reduces overfitting due to ignorance of spatial content. These two are the building block for the proposed feature enhancement network. The proposed feature enhancement network is also referred as Mixed Convolution Network due to mixing of normal and separable convolutions.

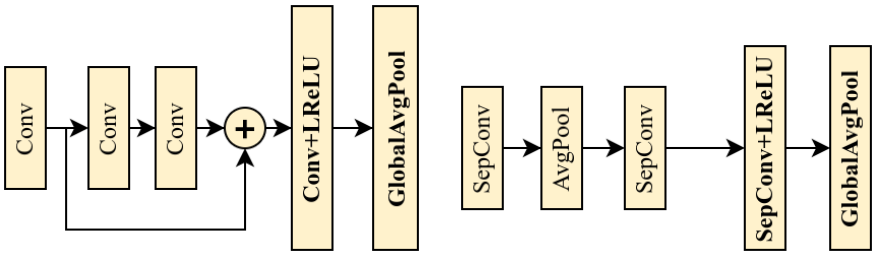

Figure 2: The proposed blocks of feature enhancement network; (a) Residual Convolution Block ; (b) Separable Convolution Block.

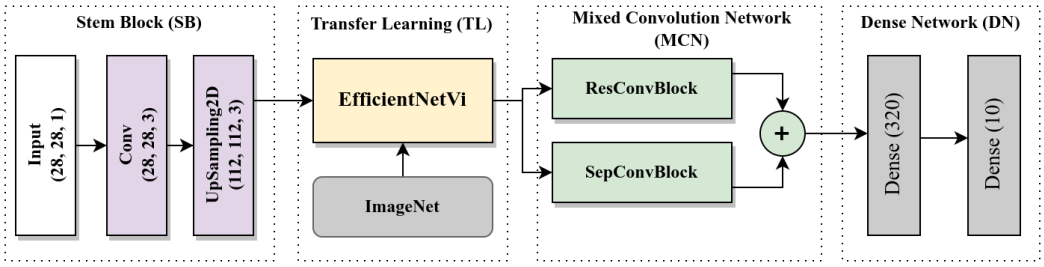

Figure 3: Deep Transfer Learning with Mixed Convolution Network.

2.2 THE PROPOSED DEEP TRANSFER LEARNING NETWORK

The design procedure for the proposed Deep Transfer Learning with Mixed Convolution Network (DTLMCN) has discussed in this section. The proposed network consists of three networks as shown in figure 3. The details of the proposed framework are as follows.

- **Feature Extraction Network**:
  This feature extraction network performs the transformation of input image I to corresponding feature map F through stem block and pre-trained network. Let us consider, the input image $I(w, h, c)$ having three dimensions first two represent the size of the image, and the third represents the number of channels. The original dimensions of the input image are (28, 28, 1), and hence it needs scaling of dimensions. The stem block (SB) has been used to perform the required scaling operation with convolution (C) and up-sampling (US) operations. The equation 1 represents the stem block operations which results in scaled image I'.

$$I' = US(C(I)) \tag{1}$$

The resulting dimension of I' is (112, 112, 3), which is compatible with the pre-trained networks. If $EBi(.)$ is the sequence of layers used by the $EfficientNetViBj$ network whose weights are initialized with Imagenet. Then, the corresponding feature map $F(w, h, c)$ can be obtained using equation 2. The output dimensions of the feature map depend on $EBi(.)$. This network captures the flavor of generalized features with transfer learning.

$$F = EBi(I') \tag{2}$$

- **Feature Enhancement Network**:
  The major contribution of the work is the ensembling of feature enhancement networks with a pre-trained network. This network is initialized with a default weight glorot initializer, unlike a pre-trained network. This network learns pure domain-specific features, and hence the proposed model blends both generalized and domain-specific features. Thus, the resulting deep transfer learning model exhibits improved performance than its base models. The equation 3 represents the sequence of operations that need to be performed in the feature enhancement network.

$$\begin{gathered} F1 = RCB(I') \\ F2 = SCB(I') \\ F = F1 + F2 \end{gathered} \tag{3}$$

- **Dense Network**:
  This network is also known as a fully connected dense network and used to produce classification results. The proposed model uses two dense layers utilized in this network as shown in figure 3 (c). The first dense layer contains 320 neurons, and the second dense layer is the output layer with 10 neurons for ten-class classification. The equation 4 represents the mapping of feature map F to class-wise predictions C.

$$\begin{gathered} D = Dense_{ReLU}(F) \\ C = Dense_{Softmax}(D) \end{gathered} \tag{4}$$

2.3 PROPOSED FUSION MODEL

Recently, the fusion models are getting attracted by researchers as they are producing a significant hike in results. Thus, the proposed work has utilized a fusion of six EfficientNet-based deep transfer learning models. The fusion model has reported a substantial improvement in performance. If $EffViBj(.)$ represents the proposed ensemble model, then algorithm 1 gives the detailed steps of the fusion model. It uses trained ensemble models to compute class prediction probabilities. Then, the class probabilities of six considered models are added together for finding the class label of the given input image.

Algorithm 1 Fusion algorithm

Input: $I(w, h, c)$ is an input image
Output: CL is predicted class label of I

1: $C_{B3} \leftarrow EffV1B3(I)$
2: $C_{B5} \leftarrow EffV1B5(I)$
3: $C_{B7} \leftarrow EffV1B7(I)$
4: $C_{v2s} \leftarrow EffV2S(I)$
5: $C_{v2m} \leftarrow EffV2M(I)$
6: $C_{v2l} \leftarrow EffV2L(I)$
7: $FC = C_{B3} + C_{B5} + C_{B7} + C_{v2s} + C_{v2m} + C_{v2l}$
8: $CL = ArgMax(FC)$

3 RESULTS AND DISCUSSION

The proposed fusion model has designed and fine-tuned for the Fashion-MNIST bench mark dataset. The dataset consists of gray images having single channel with a resolution of (28, 28). The proposed

Table 1: Performance of the proposed model

Model	Accuracy
EfficientNetB3 + MCN	95.28
EfficientNetB5 + MCN	95.43
EfficientNetB7 + MCN	95.42
EfficientNetV2S + MCN	95.15
EfficientNetVM + MCN	94.84
EfficientNetVL + MCN	95.09
Proposed fusion Model	**96.04**

Table 2: Comparison of results

Reference	Approach	Accuracy
Tanveer et al. (2021)	Fine-Tuning DARTS	96.91
Liu et al. (2018)	DARTS(2nd order) + cutout + random erasing	96.57
Foret et al. (2020)	SAM	96.41
Harris et al. (2020)	ResNet + FMix	96.36
Zhong et al. (2017)	WRN-28-10 + random erasing	96.35
	ResNeXt-8-64 + random erasing	96.21
Proposed	Fusion of DTLMCN	96.04
Phong & Ribeiro (2020)	AVG-Softmax	95.92
	EXT-Softmax	95.91
	E2E-3M	95.85
Nøkland & Eidnes (2019)	VGG8B	95.47
Rajasegaran et al. (2019)	DeepCaps	94.46
Jeevan et al. (2022)	WaveMix-Lite	94.32
Sun et al. (2020)	Neupde	92.40
Zinchenko & Lishudi (2022)	Classic Star (no warm-up)	91.30

network uses a stem block to scale the dimensions for the effective training process. The train and test dataset have taken as it is from the Keras datasets (ker, 2021). The sparse categorical cross-entropy has utilized while training the proposed models as it is the common loss function for multi-class classification. The optimizer plays a vital role in learning practice, and hence Adam optimizer with decay scheduler has employed in training. The reason behind the use of Adam optimizer is quite common to achieve a fast learning rate. The optimizer has started with an initial learning rate of 0.0005. The scheduler with a decay rate of 0.3 and decay steps of 2 has incorporated with the optimizer. The prime classification metric Accuracy has used for the validation and test performance of the proposed model.

3.1 Performance of the proposed model

Initially, the performance of the proposed fusion models have analyzed. It was observed that three of the EfficientNet version 1 models including B3, B5, and B7 have reported the best performance with MCN. Similarly, three models of version 2 with MCN have also considered for the fusion model. These six individual deep transfer learning models have reported 95% accuracy and are listed in Table 1. The accuracy has further improved to 96.04% with the fusion model. It clearly shows that there is a significant improvement of 1% than its base models.

3.2 Results analysis and discussion

From the literature, it is observed that there are several proposals have reported on the Fashion-MNIST dataset. Table 2 compares the accuracy of proposed model with existing contemporary state-of-the-art-models. The proposed fusion model exhibits competing performance with existing models with 96.04%. The pre-trained model starts with weights of Imagenet and is generalized in nature. Later, in re-training the model with the Fashion-MNIST dataset, they are slightly alighted to the

dataset. On the other hand, the feature enhancement network starts with random weights and is fine-tuned to the dataset. Thus, this network completely acquires the domain knowledge of the dataset. This makes the final model step towards the Fashion-MNIST to give the best performance. With this approach, very less training is sufficient to achieve the goal. Each of the models has attained the best performance within ten epochs only. Then, the fusion model uplifted the performance due to the addition of class probabilities.

4 CONCLUSIONS

The multi-class classification is the primary task in computer vision applications. The existing models has employed designing of new models, optimization of training parameters and enriched data augmentation techniques. The proposed work utilized pre-trained models associated with feature enhancement network. The mixed convolution network has considered for the feature enhancement in addition to pre-trained networks. This network consists of two convolution blocks including residual convolution and separable convolution. The deep transfer network network blends both generalized and domain-specific features. In addition to that, a fusion of six EfficientNet-based model has used to uplift the overall performance. The popular Fashion-MNIST dataset has considered for the experiments and the proposed model has attained 96.04% accuracy on test data. The proposed model has reported 1% improvement than its base models and exhibited competitive performance with state-of-the-art models.

REFERENCES

Fashion mnist dataset. 2021. https://keras.io/api/datasets/.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020.

Ethan Harris, Antonia Marcu, Matthew Painter, Mahesan Niranjan, Adam Prügel-Bennett, and Jonathon Hare. Fmix: Enhancing mixed sample data augmentation. *arXiv preprint arXiv:2002.12047*, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

Pranav Jeevan, Kavitha Viswanathan, and Amit Sethi. Wavemix-lite: A resource-efficient neural network for image analysis. *arXiv preprint arXiv:2205.14375*, 2022.

Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018.

Xiaobo Liu, Qiubo Hu, Yaoming Cai, and Zhihua Cai. Extreme learning machine-based ensemble transfer learning for hyperspectral image classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13:3892–3902, 2020.

Arild Nøkland and Lars Hiller Eidnes. Training neural networks with local error signals. In *International conference on machine learning*, pp. 4839–4850. PMLR, 2019.

Nguyen Huu Phong and Bernardete Ribeiro. Rethinking recurrent neural networks and other improvements for image classification. *arXiv preprint arXiv:2007.15161*, 2020.

Jathushan Rajasegaran, Vinoj Jayasundara, Sandaru Jayasekara, Hirunima Jayasekara, Suranga Seneviratne, and Ranga Rodrigo. Deepcaps: Going deeper with capsule networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10725–10733, 2019.

Afshar Shamsi, Hamzeh Asgharnezhad, Shirin Shamsi Jokandan, Abbas Khosravi, Parham M Kebria, Darius Nahavandi, Saeid Nahavandi, and Dipti Srinivasan. An uncertainty-aware transfer learning-based framework for covid-19 diagnosis. *IEEE transactions on neural networks and learning systems*, 32(4):1408–1417, 2021.

Yifan Sun, Linan Zhang, and Hayden Schaeffer. Neupde: Neural network based ordinary and partial differential equations for modeling time-dependent data. In *Mathematical and Scientific Machine Learning*, pp. 352–372. PMLR, 2020.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. 2015. *arXiv preprint arXiv:1512.00567*, 2015.

Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning (2016). *arXiv preprint arXiv:1602.07261*, 2016.

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pp. 6105–6114. PMLR, 2019.

Muhammad Suhaib Tanveer, Muhammad Umar Karim Khan, and Chong-Min Kyung. Fine-tuning darts for image classification. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 4789–4796. IEEE, 2021.

Shanshan Wang, Lei Zhang, Wangmeng Zuo, and Bob Zhang. Class-specific reconstruction transfer learning for visual recognition across domains. *IEEE Transactions on Image Processing*, 29: 2424–2438, 2019.

Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. arxiv. *arXiv preprint arXiv:1708.04896*, 2017.

Sergey Zinchenko and Dmitry Lishudi. Star algorithm for nn ensembling. *arXiv preprint arXiv:2206.00255*, 2022.