# Transfer Learning for Feature Dimensionality Reduction

Nikhila Thribhuvan
Department of Information Technology,
Rajagiri School of Engineering and Technology, India
nikhilatb@rajagiritech.edu.in

Sudheep Elayidom
Division of Computer Science, School of Engineering,
Cochin University of Science and Technology, India
sudheep@cusat.ac.in

**Abstract:** *Transfer learning is a machine learning methodology by which a model developed for achieving a task is exploited for another related job. Many pre-trained image classification models trained on ImageNet are used for transfer learning. These pre-trained networks could also be used for classifying out of domain images by retraining them. This paper, along with the existing application for these pre-trained models, is also being exploited for feature dimensionality reduction. Many dimensionality reduction methods are available; the pre-trained image models will help us perform both image feature extraction and dimensionality reduction in a single go using the same network. The fine-tuning of the fully connected layers of the pre-trained network is done to extract the image features; along with this fine-tuning, some more tweaking is done on the fully connected layers of these models to reduce the image feature dimensionality. Here, VGG-16 and VGG-19 are the pre-trained models considered for feature vector generation and dimensionality reduction. An analysis of the efficiency of features generated by these pre-trained networks in classifying the out-of-domain images is done. Three different variants of VGG-16 and VGG-19 are analysed. All the three variants developed gave an AUC value above 0.8, which is considered good.*

**Keywords:** *Dimensionality reduction, fine-tuning, transfer learning, VGG-16, VGG-19.*

*Received August 10, 2020; accepted October 13, 2021*
*https://doi.org/10.34028/iajit/19/5/3*

## 1. Introduction

Transfer learning [20] could be defined as a machine learning methodology by which a model developed for achieving a task is exploited for another related job. Transfer learning can also be thought of as an optimisation methodology for domain adaptation. In transfer learning, a base network is trained on a publicly available dataset, and this model is repurposed to utilise the learned features or transfer them to a second target network. Transfer learning is used chiefly to perform predictive modelling problems that use image data as input and natural language processing problems that use text as input or output.

One advantage of transfer learning is that it is an optimisation technique and a shortcut to save time or improve performance. Another advantage is when there is a lack of training data. These trained models can be utilised to build customised models during the scarcity of data is there. At times, it is challenging to choose which would be the best pre-trained model to be selected. That may require assistance from the domain experts or should go for a brute force approach. Examples of such models include Oxford's VGG model, Google's Inception model, Microsoft's ResNet Model, DenseNet, NASNet, [6] etc., The organisations that develop these models release these models with the reuse license for research and commercial purposes.

All the above mentioned pre-trained models follow CNN architecture [17]. The pre-trained Convolutional Neural Networks (CNNs) are the best suited deep learning models that are used for image classification [14] and feature extraction because the pre-processing and training required is much lower when compared with other image classification algorithms. However, training a CNN from scratch is a herculean task as most of the CNN requires a tremendous amount of training data and requires much time to prepare the network. This paper deals with the pre-trained CNN model, VGG-16 and VGG-19, and how they are used to extract the image features of out of domain images along with their feature dimensionality reduction.

The pre-trained CNN models can be used for three primary purposes:

1. Image classification [7]
2. Standalone and integrated feature extraction
3. Image feature extraction of out of domain data
4. Feature dimensionality reduction

The integrated feature means some portion of the pre-trained network could be integrated into another model, freezing some model layers. The major contribution of this work could be marked in the field of dimensionality reduction, that is, the use of fine-tuned pre-trained networks for feature vector dimensionality reduction.

Even though dimensionality reduction will result in the loss of some percentage of information, dimensionality reduction is a pre-processing step [5] before training a model as it will lead to certain advantages like:

- Lesser training time and less use of computational resources.
- Reduces the complexity of the model and avoids the problem of overfitting.
- Provides better data visualisation.
- Addresses the problem of multicollinearity.
- Improves the model accuracy by retaining the most critical features.

The following section explains areas where the pre-trained models generate the out-of-domain features and how dimensionality can be reduced in the extracted feature. Next, the section, fine-tuning of pre-trained networks, explains in detail the architecture used and then presents different transfer learning models and their efficacy in classification.

## 2. Related Works

Pre-trained [6] models are models created by some other people for solving similar problems. These models could be reused instead of building a new model from scratch. These models will not give 100% accuracy but save a huge lot of time and effort. VGG is a pre-trained network trained over ImageNet [8, 17]. The model achieves a test accuracy of 92.7% on ImageNet, which has over 14 million images belonging to 1000 different classes. There are two variants of the VGG model, VGG-16 and VGG-19.VGG-16 [21] is a 16 layered architecture with 16 layers, out of which 13 are convolutional layers, and 3 are fully connected layers. In the case of VGG-19, there will be 19 layers, out of which 16 are convolutional, and 3 are fully connected layers. These pre-trained models are used in the areas of sketch recognition, human activity recognition, [19, 23] etc., that can improve the efficiency of training required.

At times there can be a situation when there is a requirement for a reduced feature dimension for reducing the complexity of computation. After extracting the feature vectors, some dimensionality reduction techniques like Principal Component Analysis (PCA) [1], linear discriminant supervised laplacian eigenmaps or supervised locality preserving projection can be used. The problem with all the dimensionality reduction methods mentioned above is that they allow only linear data transformation. On the other hand, deep learning models' visible and hidden units can learn more complicated relations between the data [22]. Using deep learning models, correlations between the features and the target can be visualised. These correlations help in removing features that do not affect the target. For example, out of 50 features in the dataset, if 90 percent of the target variable is affected by ten of the features, the others could be shaved off, making the machine learning model a lot simpler. Here, instead of applying another methodology for dimensionality reduction, the existing model is finely tuned to get a reduced feature vector.

An American e-commerce website, Etsy, used the concept of multimodality based search ranking using deep Convolutional networks to improve the quality of search results [10]. In the paper, Lynch *et al*. [10] proposes how visual information in images can be used along with its textual descriptor for ranking the results to the user. The high-level visual details in the images are extracted using a deep Convolutional Network using transfer learning. The pre-trained VGG-19 is used to extract the feature vector of the image that is combined with the existing ranking method to provide a better search experience. The 4096 feature vector of the image is being extracted from the fully connected layers of VGG-19.

When there are not enough labelled data to train the model, the representations learned by CNN for different domain data could be helpful. These networks could also be used for out of domain feature extraction [11]. The CNN layers and final layers of AlexNet architecture are fine-tuned with plankton dataset images. The features of the plankton images are pulled from the last fully connected layer.

Pre-trained CNN models are even used in the field of Facial Expression Recognition (FER). However, the conventional FER may not operate well as they are trained against constrained real-time datasets, limiting the accuracy and efficacy of these systems. To overcome this, the pre-trained models are used to extract the spatial features and combined with other models like LSTM to extract the temporal features for FER [13]. Here, the concept of transfer learning is used in extracting the features of out of domain images for which the training data is sparse.

After extracting the feature vectors, some dimensionality reduction techniques like PCA, Linear Discriminant Supervised Laplacian Eigenmaps, Supervised Locality Preserving Projection, etc. can be used. Cascianelli *et al.* [4] made feature dimensionality reduction on the features extracted from different architectures of VGG. The features from the fully connected layers were extracted, and separate dimensionality reduction methods like PCA, Gaussian Random Projection (GRP) and Correlation-based Feature Selection (CBFS) were applied. To minimise the variance of the projected data in the first two methods, the data is projected to a lower dimension space. CBFS, reduces the dimensionality by retaining only those features whose cross-correlation is beyond a threshold value. The CBFS method was the best dimensionality reduction method because it includes only those features with more discriminating characteristics.

The basic principle of a neural network is that the neurons will learn the most discriminating features [22]. So, instead of applying another methodology for dimensionality reduction, the existing CNN model itself is finely tuned by reducing the number of neurons in the fully connected layers to get a reduced dimensionality feature vector. The aim is met by replacing some of the layers of the existing pre-trained model and retraining them.

# 3. Fine-tuning of Pre-Trained Networks

## 3.1. Feature Extraction-Proposed Architecture

VGG-16 and VGG-19 are used for feature extraction. The convolutional layers of the pre-trained models are initialised with weights based on training on the ImageNet dataset [3]. ImageNet has 1000 classes; the whole network could be trained with these extra classes for the model to classify more classes. The convolutional layers can be frozen to keep the same feature extractors. The final Softmax layer responsible for classification can be removed from the model, and the prefinal layer can be considered for providing the feature vector as in Figure 1. Another method is extracting the feature from the last maxpool layer after the 13$^{th}$ convolutional layer. This output is flattened and passed to some dimensionality reduction method.

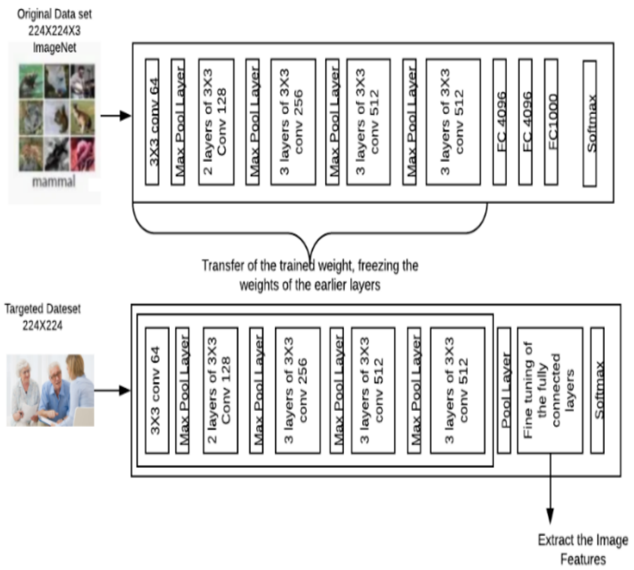


Figure 1. Transfer learning: feature extraction- proposed architecture.

Here, the first method of feature extraction is being adopted. For extracting image features, the Fully Connected (FC) layers are fine-tuned [4], and the max pool [1] layer is replaced with the Global average layer, as shown in Figure 2. Pooling layers reduce the variance and computational complexity in neural networks. Maxpool layers extract the most prominent or important feature, whereas, global average pool layer sums up the spatial information in an image. Thus, it reduces the possibility of overfitting also.

The fully connected layers of VGG extract 4096 features of the image. To reduce the dimensionality of the extracted feature vector, the shape of the FC layers is modified and the model is fine-tuned using 32 and 128 selected classes of ImageNet. All the other layers are frozen till block5_conv3 (refer Figure 2), that is, till the 13th convolutional layer [15].

The fine-tuned model minimises the cross-entropy loss function using the Adagrad [16] optimiser algorithm. It is an adaptive second-order type optimisation algorithm that changes the learning rate ‘η’ for each parameter and step ‘t’. It works on the derivative of an error function. It adapts the learning rate of the parameters, performing larger updates for infrequent and smaller updates for frequent parameters. Dropout layers are also introduced in between the fully connected layers to avoid overfitting. The model is fine-tuned to classify 32 and 128 classes from the ImageNet dataset. The model summary shows that the total parameters of the model are 14, 785, 536, and the number of trainable parameters is 70, 848. It clearly shows that the number of trainable parameters is much less when compared with the total number of parameters present. This reduction in the trainable parameters is the main advantage of the process of transfer learning, which in turn, will reduce the training time taken.

When fine-tuning the fully connected layers, the network is checked for over fitting and under fitting [18]. The model can overlearn or under learn the features, which can result in over fitting or under fitting of the model. Over fitting occurs when the model learns even the noise of the dataset, with low bias and high variance. Under fitting [2] occurs when the model cannot capture the underlying trend in the dataset. Both of them result in a wrong prediction. Any network with many trainable features suffers mainly from the problem of over fitting that can be checked for using the testing and training loss-accuracy graph. Learning too many features reduces the generalisation capability of the model. Over fitting [2] can be avoided by:

1. Reducing the layers in the model.
2. Early stopping [12].
3. Data Augmentation.
4. Regularisation (by modifying the cost function)
5. Adding dropouts (by randomly dropping neurons during the different iteration of the training process).

The method adopted here to overcome the problem of overfitting is adding dropout layers. As the name implies, the neurons are being dropped out temporarily along with their inputs and outputs. Therefore, two dropout layers, each with a dropout rate of 0.25, are added to the model.

```
Layer (type)                 Output Shape              Param #
=================================================================
block1_conv1 (Conv2D)        (None, 224, 224, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 224, 224, 64)      36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 112, 112, 64)      0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 112, 112, 128)     73856
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 112, 112, 128)     147584
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 56, 56, 128)       0
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 56, 56, 256)       295168
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 28, 28, 256)       0
_________________________________________________________________
block4_conv1 (Conv2D)        (None, 28, 28, 512)       1180160
_________________________________________________________________
block4_conv2 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_conv3 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 14, 14, 512)       0
_________________________________________________________________
block5_conv1 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_conv2 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_conv3 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_pool (GlobalAveragePo (None, 512)               0
_________________________________________________________________
fc1 (Dense)                  (None, 128)               65664
_________________________________________________________________
dropout_3 (Dropout)          (None, 128)               0
_________________________________________________________________
fc2 (Dense)                  (None, 32)                4128
_________________________________________________________________
dropout_4 (Dropout)          (None, 32)                0
_________________________________________________________________
predictions (Dense)          (None, 32)                1056
=================================================================
Total params: 14,785,536
Trainable params: 70,848
Non-trainable params: 14,714,688
```

Figure 2. Model summary of fine-tuned VGG-16 model.

### 3.2. Dimensionality Reduction

The transformation of data from a high dimensional space to a lower-dimensional space is called dimensionality reduction [14]. This can be done by using two methods: feature selection and feature projection or feature extraction. In feature selection, the subset of the input is formed, whereas, in feature projection, data in a higher dimension is converted to a low extent. Feature projection for dimensionality reduction is adopted here. Feature projection can be either linear or non-linear. Dimensionality reduction can be used for noise reduction also. It can even be used for cluster analysis and data visualisation. In machine learning, it is generally said that the more the data, the more precise the model will be, and they will have more generalisation capability. But more noise will affect the training time of a model.

Dimensionality reduction might be needed when combining vectors from two different datasets to train a new model. While using deep learning models to reduce dimensionality, the model tries to avoid the less important features. The dropout layers implemented between the fully connected layers also help drop out these kinds of feature values. The fully connected layers are retrained using subclasses from ImageNet. The set of models that are tried out is given below.

All the pre-trained VGG models are trained to provide a 4096-dimension feature vector. At times this dimension of the feature vector might be too much, where the dimensionality is reduced. Here, the fully connected layers of VGG-16 are reshaped, as seen in Figure 2 for TL model 1. The first fully connected layer has 128 neurons, and the second one is with 32 neurons. These two fully connected layers are retrained using 32 subclasses of ImageNet. The same is repeated for VGG-16 with 128 classes and VGG-19 with 128 classes. The accuracy in classification with this reduced feature vector is evaluated to analyse which is the best model in feature dimensionality reduction VGG-19 is classified as the best model for the same.

Three different transfer learning models tried are:

1) TL Model 1-Using VGG-16, fine-tuned over 32 classes.
2) TL Model 2-Using VGG-16, fine-tuned over 128 classes.
3) TL Model 3-Using VGG-19, fine-tuned over 128 classes.

## 4. Model Performance Evaluation

The model is fine-tuned using Keras framework [2] and different measures of the model evaluation are plotted as graphs below. The model is evaluated using the loss accuracy learning curve. From the given graph (Figure 3), it could be seen that the TL model 1 could attain an accuracy of 0.82 and loss of 0.54 at 30 epochs of training. The test accuracy of the model is 0.74, and the loss is 0.5. The model loss graph is shown in Figure 4. The AUC value attained is 0.83 is achieved for TL Model 1,0.8 is achieved for TL Model 2 and 0.85 is achieved for TL Model 3. An AUC value of 0.8 and above shows that the models has got good discriminating or classification capability even with the reduced feature dimension. It can be seen that the number of trainable parameters has also come down considerably from Figure 2.

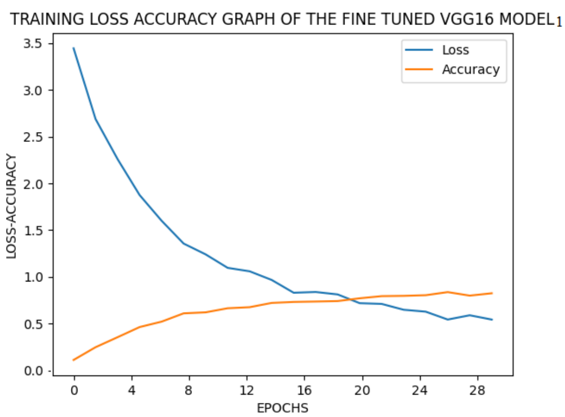


Figure 3. Loss-accuracy graph of the fine-tuned TL model 1.

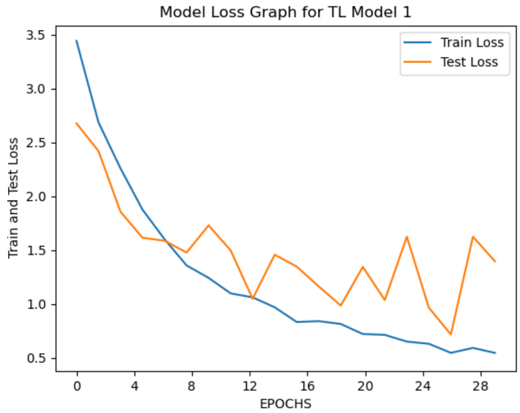


Figure 4. Model loss of fine-tuned TL model 1.

Table 1. Loss-Accuracy of 3 transfer learning models.

| Transfer Learning Models | Train Accuracy | Train Loss | Test Accuracy | Test Loss |
|---|---|---|---|---|
| TL Model 1- Using VGG-16 | .82 | .5 | .74 | 1.3 |
| TL Model 2- Using VGG-16 | .73 | .9 | .7 | 1.4 |
| TL Model 3- Using VGG-19 | .73 | .9 | .69 | 1.2 |

It is clear from the graph, Figure 4 that the loss graph for training and testing is not converging with each other, they are getting separated with the increase in their difference, and it is an indication of the overlearning on the training samples. To upgrade the transfer learning performance, two more models were tried, which are VGG-16 and VGG-19 fine-tuned on 128 classes. Their loss and accuracy values are listed in Table 1. The corresponding loss- accuracy graphs and model loss are also plotted (Figure 5). It is seen that the VGG-16 trained in 128 class and VGG-19 trained in 128 class are exhibiting almost the same performance. However, slightly better performance could be noted for VGG-19 as the test and train values converge much more sooner. The test loss for VGG -16 is 1.4, and for VGG-19 it is 1.2 even though all the other values are almost similar.

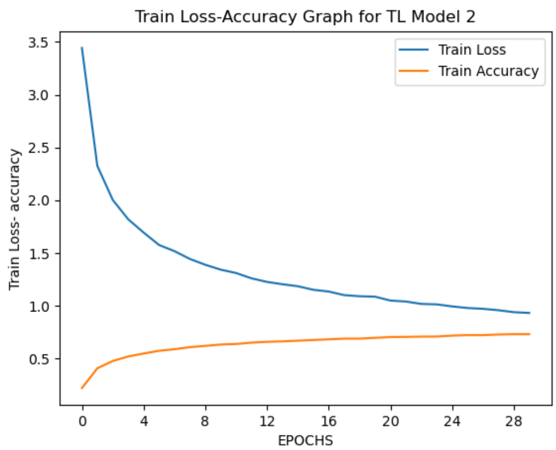


Figure 5. Train loss-accuracy graph of the fine-tuned TL model 2.

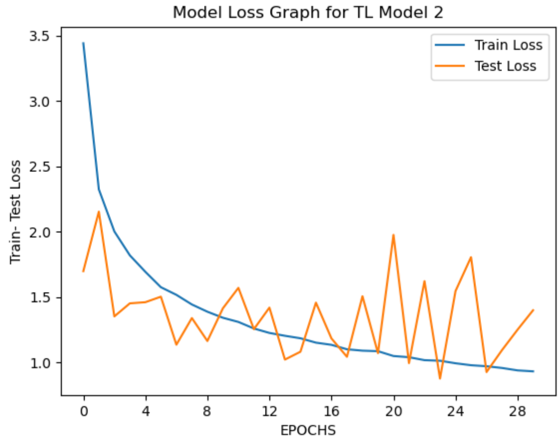


Figure 6. Model loss of fine-tuned TL model 2.

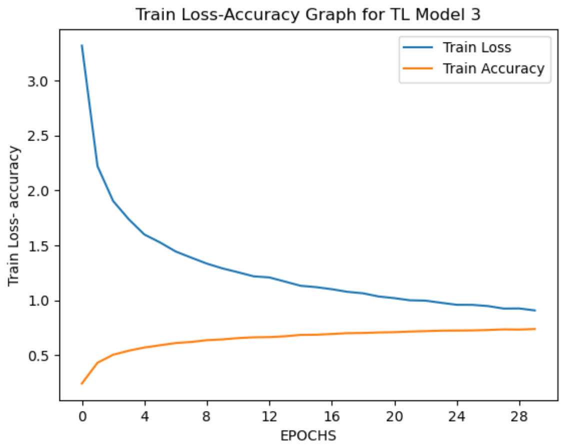


Figure 7. Loss-Accuracy graph of the fine-tuned TL model 3.

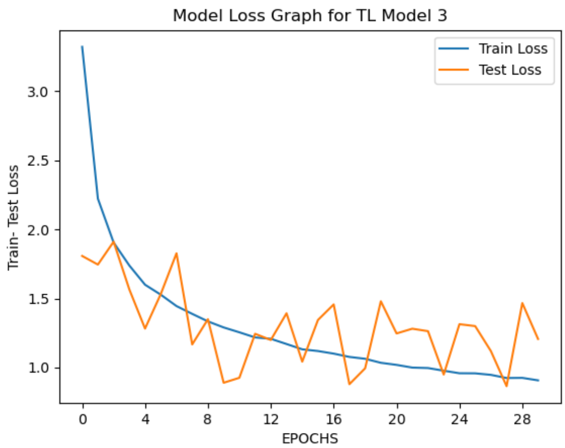


Figure 8. Model loss of fine-tuned TL model 3.

From the Table 1, it is clear that even though the training accuracy and loss of TL model 1 (Figure 3) gave a better value, it considerably went down during the testing phase. The model might have overlearned the features in the training dataset. The model loss graph of TL model 1 also shows the same as the two graphs do not converge. The model loss graph of TL model 2 (Figure 6) and TL model 3 (Figure 8) shows that the training and testing loss graphs are converging after around 10 epochs even though there is slight fluctuation. However, VGG-16 and VGG-19 trained in 128 classes are giving a similar performance (Figures 5 and 7) and VGG-19 could be nominated as the best for feature extraction and dimensionality reduction as it gives less test loss. The same models are also tested for out of domain images (MS COCO) and provided similar classification accuracies.

## 5. Conclusions

All the dimensionality reduction methods can perform the reduction of features given the feature vector. This architecture helps us to extract features as well as reduce the dimensionality in a single go. This is the main highlight of the method discussed. More fully connected layers can be added to get more precise features, increasing the training time. The accuracy can be improved while new fully connected layers are added, but the overhead in training overweighs the advantage. So, it is better to restrict the layers up to 2 or 3, which gives a considerable accuracy in classification, as shown in the table above.

Even though there are many dimensionality reduction methodologies available, when working with pre-trained models for feature extraction, it would be better to use them for dimensionality reduction. The use of another new method increases the overhead of the model. The deep learning models are best suited to learn the more complicated relationship between the features. In future, this method could be compared with other existing dimensionality reduction methodologies also. The reduced features extracted using this method can be used as feature vectors for many Learning to Rank models [9]. These features extracted from pre-trained models are used as input for other models in the field of urban and regional study, facial recognition, object identification, lesion detection, etc.

## Reference


[1] Alkhayrat M., Aljnidi M., and Aljoumaa K., "A Comparative Dimensionality Reduction Study in Telecom Customer Segmentation Using Deep Learning and PCA," *Journal of Big Data*, vol. 7, no. 9, pp. 2-23, 2020.

[2] Astion M., Wener M., Thomas R., Hunder G., and Bloch D., "Overtraining in Neural Networks that Interpret Clinical Data," *Clinical Chemistry*, vol. 39, no. 9, pp.1998-2004, 1993.

[3] Brownlee J., A Gentle Introduction to Pooling Layers for Convolutional Neural Networks, https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/, Last Visited, 2019.

[4] Cascianelli S., Bello-Cerezo R., Bianconi F., Fravolini M., Belal M., Palumbo B., and Kather J., "Dimensionality Reduction Strategies for CNN-Based Classification of Histopathological Images," *in Proceedings of Intelligent Interactive Multimedia Systems and Services*, Gold Coast, pp. 21-30, 2017.

[5] ChandraSekhar B., Niranjan A., and Reddy G., "Dimensionality Reduction using Deep Learning Techniques," *International Journal of Psychosocial Rehabilitation,* vol. 24, no. 6, pp. 1137-1143, 2020.

[6] Karim R., "Illustrated: 10 CNN Architectures:A Compiled Visualisation Of The Common Convolutional Neural Networks, https://towardsdatascience.com /illustrated-10-cnn-architectures-95d78ace614d, Last Visited, 2019.

[7] Krishna S. and Kalluri H., "Deep Learning and Transfer Learning Approaches for Image Classification," *International Journal of Recent Technology and Engineering*, vol. 7, no. 5S4, pp. 427-432, 2019.

[8] Krizhevsky A., Sutskever I., and Hinton G., "Imagenet Classification with Deep Convolutional Neural Networks," *in Proceedings of the 25th International Conference on Neural Information Processing Systems*, USA, pp. 1097-1105, 2012.

[9] Li H., *Learning to Rank for Information Retrieval and Natural Language Processing*, Synthesis Lectures on Human Language Technologies, Morgan and Claypool, 2011.

[10] Lynch C., Aryafar K., and Attenberg J., "Images Don't Lie: Transferring Deep Visual Semantic Features to Large-Scale Multimodal Learning to Rank," *in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, pp. 393-402, 2016.

[11] Orenstein E. and Beijbom O., "Transfer Learning And Deep Feature Extraction for Planktonic Image Dataset," *in Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, Santa Rosa, pp. 1082-1088, 2017.

[12] Prechelt L., *Neural Networks: Tricks of the Trade*, Springer, 2012.

[13] Rajan S., Chenniappan P., Devaraj S., and Madian N., "Novel Deep Learning Model for Facial Expression Recognition Based on Maximum Boosted CNN and LSTM," *IET Image Process*, vol. 14, no. 7, pp. 1373-1381, 2020.

[14] Rawat W. and Wang Z., "Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review," *Neural Computation*, vol. 29, no. 9, pp. 2352-2449, 2017.

[15] Rosebrock A., "Fine-tuning with Keras and Deep Learning, https://www.pyimagesearch.com /2019/06/03/fine-tuning-with-keras-and-deep-Learning," Last Visited, 2019.

[16] Ruder S., "An Overview of Gradient Descent Optimisation Algorithms," *ArXiv* abs/1609.0474, 2016.

[17] Simonyan K. and Zisserman A., "Very Deep Convolutional Networks for Large-Scale Image Recognition," arXiv:1409.1556, 2014.

[18] Srivastava N., Hinton G., Krizhevsky A., Sutskever I., and Salakhutdinov R., "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp.1929-1958, 2014.

[19] Tenorio D., Villalobos C., Mendoza L., Silva E., and Caarls W., "Improving Transfer Learning Performance: An Application in the Classification of Remote Sensing," *in Proceedings of the 11th International Conference on Agents and Artificial Intelligence*, Prague, 2019.

[20] Torrey L. and Shavlik J., *Transfer Learning in Handbook of Research on Machine Learning Applications*, IGI Global, 2009.

[21] VGG16 and VGG19, https://keras.io/api/applications/vgg/, Last Visited, 2019.

[22] Yann L., Haffner P., Bottou L., and Bengio Y., *Shape, Contour and Grouping in Computer Vision. Lecture Notes in Computer Science*, Springer, 1999.

[23] Zhou W. and Jia J., "Training Convolutional Neural Network for Sketch Recognition on Large-Scale Dataset," *The International Arab Journal of Information Technology*, vol. 17, no. 1, pp. 82-89, 2020.

**Nikhila Thribhuvan** She is working as an Assistant professor in the Department of Information Technology at Rajagiri School of Engineering & Technology, Kakanad, India. She was awarded PhD in faculty of engineering(CS)for the topic " Design and Implementation of a multimodal Learning to Rank model to rank web pages " from Cochin University of Science and Technology, Kerala, Kochi, India. She completed her B.Tech from the Kannur University and M.Tech from IGNOU. She has got 12 years of teaching experience. She is the chapter author for the book, "Data Mining and Warehousing". She has a patent published under the application No.: 202141042022 for the title IoT Enabled Step to Ramp for Mobility Impaired. Her areas of Interest are Cloud Computing, Information Retrieval, AI and Machine Learning.

**Sudheep Elayidom** He is a computer engineer with a PhD in data mining. Currently he is working as a professor and PhD guide at School of Engineering, CUSAT, Kochi, India. He completed his B.Tech from the M.G University and M.Tech from CUSAT, with first ranks in both. He has several international publications to his credit. His areas of interest include data mining, databases, software engineering, object oriented analysis, Big Data, Cloud databases, etc. He has more than 20 years of teaching experience and is the doctoral committee member of many researchers. He is the author of the book by the name "Data Mining and Warehousing", which is a syllabus prescribed text book for courses on Data Mining in many universities across the country