symmetry

MDPI

Article

Lifelong Machine Learning for Regional-Based Image Classification in Open Datasets

Hashem Alyami [1,*], Abdullah Alharbi [2] and Irfan Uddin [3]

[1] Department of Computer Science, College of Computers and Information Technology, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia
[2] Department of Information Technology, College of Computers and Information Technology, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia; amharbi@tu.edu.sa
[3] Institute of Computing, Kohat University of Science and Technology, Kohat 26000, Pakistan; irfanuddin@kust.edu.pk
* Correspondence: hyami@tu.edu.sa

Received: 22 November 2020; Accepted: 13 December 2020; Published: 16 December 2020

Abstract: Deep Learning algorithms are becoming common in solving different supervised and unsupervised learning problems. Different deep learning algorithms were developed in last decade to solve different learning problems in different domains such as computer vision, speech recognition, machine translation, etc. In the research field of computer vision, it is observed that deep learning has become overwhelmingly popular. In solving computer vision related problems, we first take a CNN (Convolutional Neural Network) which is trained from scratch or some times a pre-trained model is taken and further fine-tuned based on the dataset that is available. The problem of training the model from scratch on new datasets suffers from *catastrophic forgetting*. Which means that when a new dataset is used to train the model, it forgets the knowledge it has obtained from an existing dataset. In other words different datasets does not help the model to increase its knowledge. The problem with the pre-trained models is that mostly CNN models are trained on open datasets, where the data set contains instances from specific regions. This results into predicting disturbing labels when the same model is used for instances of datasets collected in a different region. Therefore, there is a need to find a solution on how to reduce the gap of Geo-diversity in different computer vision problems in developing world. In this paper, we explore the problems of models that were trained from scratch along with models which are pre-trained on a large dataset, using a dataset specifically developed to understand the geo-diversity issues in open datasets. The dataset contains images of different wedding scenarios in South Asian countries. We developed a Lifelong CNN that can incrementally increase knowledge i.e., the CNN learns labels from the new dataset but includes the existing knowledge of open data sets. The proposed model demonstrates highest accuracy compared to models trained from scratch or pre-trained model.

Keywords: lifelong machine learning; deep learning; convolutional neural network; computer vision

1. Introduction

Over the last decade, it was observed that pattern recognitions systems perform more efficiently when we use Neural Network (NN) or particularly Deep Neural Networks (DNNs). Recent development in deep learning has revolutionized the field, particularly the developments in CNN. These techniques heavily depend on the data the models are trained. However, it has been observed by different researchers [1] that deep learning algorithms can give incorrect results or some times the results are disturbing when there is not enough representation of the data belonged to a particular class or region. For instance, irrigation-based techniques or equipments, work very well in certain crops

or areas but do not work well when the system is used in a different environment. Another example could be self-driving cars, where cars can drive very efficiently on asphalt roads, but the chances of failure or accidents increase when there are dirt or gravel. These deep learning based techniques are as good as the data they learn from. To build deep learning systems that can work efficiently for the whole world in different diverse regions, the training data must be designed that have representations from diverse regions or there need to be improvements in the algorithms of learning.

CNN models have been efficiently and effectively used for solving problems in pattern recognition or image recognition such as classification of objects [2], face recognition [3], gesture recognition [4] and image captioning [5]. ImageNet [6] and Open Images datasets are commonly used to develop systems for image classification or image recognition problems. ImageNet dataset consists of approximately 15 million images from 21,841 different categories. However, most of the research papers use 1.2 million images of 1000 different categories, and hence most researchers reports accuracy of networks where their models are trained with ImageNet consisting of 1.2 million images of 1000 classes. Images are associated with a human-verified single label. Open Images dataset contains around 9 million images annotated with labels spanning over 6000 categories. The creators of the dataset, have tried to make the dataset as practical as possible by making labels that cover more real-life entities than the 1000 ImageNet classes. These datasets contain images mostly from US, UK, Europe or Australia. There is not enough representation from developing world such as Ethiopia, India, Pakistan, Afghanistan, China, Saudi Arabia etc. The number of images in ImageNet and Open Images based on different regions/countries is analyzed in [1] and presented in Figure 1. It is clear that most of the images are taken from US, Canada, Europe and Australia. This can results into biasness in classification problems when used to make classification on images collected from a different region, because there is not enough representation from developing world as shown in Figure 2, where *bridegroom* from Pakistan and Ethiopia are not correctly classified, while *bridegroom* from US, Australia, Canada are easily classified.

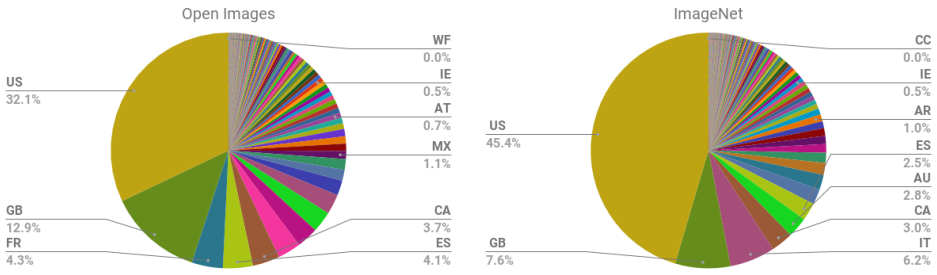


Figure 1. Fraction of images in Open Images and ImageNet datasets collected in different regions. In both datasets, majority of the images are collected from US, UK, Canada and Australia. In the image the two-letter code represents country name (Source [1]).

Humans and animals transfer the knowledge and skills learning throughout the lifespan. We have a complex neuro-cognitive system that help the sensorimotor skills to develop and specialize and at the same time develop long-term memory storage and retrieval of information [7]. This ability of transferring the knowledge and skills throughout the lifespan is referred as lifelong learning. These skills are very much important for computers and intelligent systems working in real-world environment and process the flow of information. Lifelong Machine Learning (LML) [8] involves mechanisms that can learn different tasks from different domains over the span of its lifetime. The idea of LML is to retain learned knowledge and to selectively transfer knowledge when learning a new task so as to develop more accurate hypothesis or policies. This paper further emphasizes the need for Artificial Intelligence (AI) applications to not only learn based on existing dataset, but need to understand the systems that are able to learn over a lifetime. We are presenting an algorithm for CNN that is used for transferring knowledge from ImageNet and useful for the dataset containing images in

different scenarios and in different regions. The results of this paper conclude that the next logical step in the domain of supervised learning is LML.

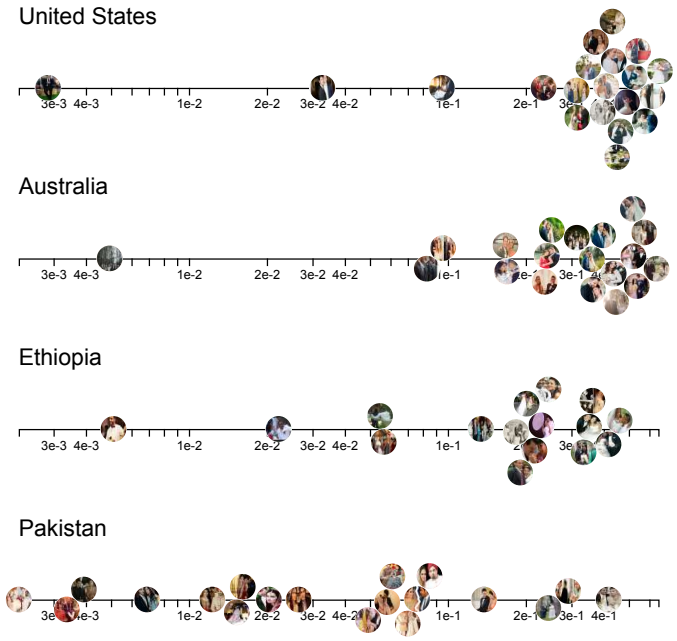


Figure 2. Photos of *bridegrooms* from different countries aligned by the log-likelihood that the classifier trained on Open Images assigns to the *bridegroom* class. Images from Ethiopia and Pakistan are not classified as accurately as images from the regions such as US, UK, Canada, Australia, where the model is trained on (Source [1]).

The organization of the paper is as follows. In Section 2 we briefly explain the current popular techniques in literature. The proposed technique is explained at Section 3. We also discuss briefly the dataset specifically prepared for this paper and explanation of the current popular models typically used for classification of images and object recognition. We further explain an algorithm that can build the knowledge based on existing knowledge. Experiments collected from models pre-trained on large dataset and models trained on new dataset from scratch and Lifelong CNN is explained in Section 4. The conclusion of the paper along with possible future directions are given in Section 5.

2. Literature Study

An analysis of different Convolutional Neural Network (CNN) models for classification of Images is given in [9]. The paper explains in detail the networks such as; LeNet5, AlexNet, ZFNet, VGGNet, GoogLeNet (also known as Inception), ResNet, DenseNet and CapsNet. The authors' claims are based on different experiments that GoogLeNet and ResNet obtained good performance rather than simply putting the building blocks of CNN in a sequential fashion. In [10] CNN is used to identify disease in plants by taking an image of the plant. To determine the health of plants through an image, it is very challenging as the crops have rich and complex environment. Authors have used AlexNet, DenseNet, Inception, ResNet, SqueezeNet and VGG. In [11] a CNN is trained to classify traffic signs. In the paper, authors have explored transfer learning techniques known as *fine tuning technique* to reuse layers pre-trained on ImageNet dataset.

In [12] an empirical analysis of performance of popular CNNs for identifying objects in real-time videos is made. The paper focuses on evaluating the performance of AlexNet, GoogLeNet and ResNet50 using datasets CIFAR10, CIFAR100 and ImageNet. The paper concludes that ResNet50 and GoogLeNet gives better performance on all three datasets. In [13], image recognition is performed using a CNN with (3×3) filters for convolution, and has demonstrated significant improvement by

obtaining good accuracy. In [14] a CNN is used to make classification of 1.2 million images collected in the ImageNet to make classification in 1000 different classes. The performance on the test data is shown as 37.5% top-1 error rate which are considered improvement over previous popular techniques. In [15] multiple tricks in CNN for accuracy improvement in image classification are explained. The paper demonstrates that better accuracy in image classification makes transfer learning more efficient in different applications such as object detection or segmentation.

In [16] a CNN architecture that shares latent factorized representations in CNNs is developed. Deconvolutional factorization along with tensor contraction are used to make a transformation among multiple operations. They have tested two datasets i.e., CIFAR-10 and 100 and results demonstrate that the developed method has obtained good performance for difficult setting of lifelong learning, avoiding the problem of catastrophic forgetting and uses backward transfer to make improvement in the performance in learning by improving previously learned tasks from different experiments without requiring to retrain. In [17], authors are explaining that though computer vision technology is used by many people are around the world but these techniques (datasets) contain representation of only a few regions, and hence it is been reported that the computer vision models misbehaves in predicting labels that are offensive or low accuracy in unrepresented regions. They have analyzed that the datasets are typically manually annotated images or videos, and the label distribution is not fair. They have analyzed ImageNet and considered three factors for the *person* category; (i) the vocabulary used for annotation (ii) exhaustive illustration of all categories (iii) inequality of representation. They had taken a first step to eliminate unfairness in ImageNet.

A Tree-CNN model is proposed in [18] for data with low frequency. Three key insights are taken into consideration: (i) energy consumption in household is based on patterns based on time and this pattern can be captured by the different kernels in CNN (ii) the structural representation as a tree allows learning the structure of individual products and therefore the difference in magnitude is avoided while retaining the relationship between appliances (iii) known and unknown appliances are separated, and therefore the input time series is better used for reconstruction of time series for appliances. Authors claim that the performance of Tree-CNN-based model is better than current popular models in terms of lower prediction error and better performance in detection of states that are active in different appliances. In [19] a new CNN is developed that consists of clustering algorithm and Tree-CNN. The role of the clustering algorithm is to make classification in a high-level class. Tree-CNN consists of Trunk-CNN and Branch-CNN. The Trunk-CNN is used for coarse classification, and the role of the Branch-CNN is to make difference between groups of same category. The authors used *Caltech101* and *Caltech256* datasets testing the model and have demonstrated superior results by Tree-CNN. Some other related studies can be found in [20–23].

The techniques presented in literature are developed for different computer vision activities. Different CNN models are developed to solve problems in object recognition, object detection, segmentation, etc. These models suffer from catastrophic forgetting. Which means that the model is able to train an existing dataset with high accuracy, but when the model is presented with a new dataset the performance of the system is low. The model is not able to train itself on the new dataset while keeping the knowledge of the previous data set. To deal with this situation, the paper has presented a novel technique that is based on the hierarchical representation of objects. Using this model we are able to train on new dataset, while maintaining the knowledge of old datasets. The results have demonstrated that the model has achieved better performance accuracy compared to other CNN models.

3. Proposed Methodology

3.1. Preprocessing Analysis

Here, we explain the dataset obtained specifically for this paper to demonstrate the geo-diversity issues in open datasets. The preprocessing on the dataset is also explained in this section. The dataset

contains images collected mainly from different regions of Pakistan, India, and Afghanistan, where each image contains a wedding scenarios. For example, images containing Brides or Grooms or cars decorated for wedding etc. The dataset is called *Wedding* dataset. Three different annotators are used to annotate images and the annotation with highest accuracy percentage is shown as final value to be shown as result. The different classes used in the Wedding dataset is given in Table 1 and a sample of images from the dataset is illustrated in Figure 3.

Table 1. The categories in the Wedding dataset showing the distribution of images in train and test datasets.

Label	Category	Train	Test
0	Bride	787	199
1	NotWeddingCar	760	164
2	Formal	720	189
3	Groom	797	205
4	NotBride	791	208
5	NotFormal	431	109
6	NotGroom	735	188
7	WeddingCar	489	116

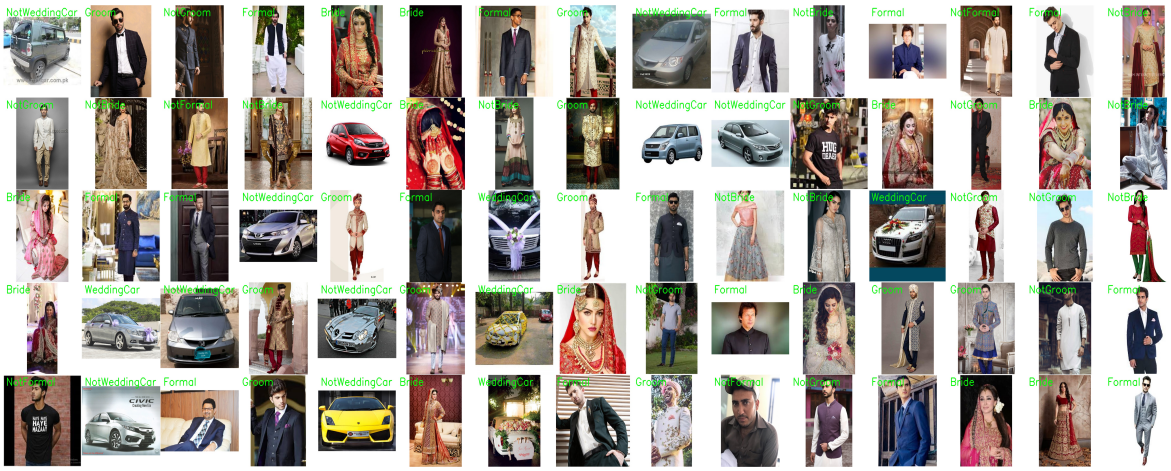


Figure 3. A sample collection of images from the Wedding dataset.

3.2. Normalization

Colored images contain pixel values between 0 and 255. Therefore, some features in the CNN have small values in the range of $[0, 10]$ while others have large values such $[250, 255]$. The ML algorithms have difficulty to learn in this situation. Therefore, the data is normalized in the range of $[0, 1]$ to be efficiently processed by ML models. In this paper, all pixel values are divided by 255 to convert the values of pixels to be in the range 0 to 1.

3.3. Image Classification Models

This section gives a brief explanation of the most popular CNN architectures commonly used for problems involving computer vision. These three architectures are; LeNet-5, ResNet50 and Inception. This section also gives explanation of Tree-CNN, that is adapted in this paper to be used for classification of wedding scenarios in Wedding dataset.

3.3.1. LeNet, ResNet and Inception

LeNet-5 [24], presented by LeCun et. al. was the exceptionally to begin with CNN show used for acknowledgment of manually written and machine-printed characters. The model has two convolution

and average pooling layers, taken after by two completely connected layers and a *softmax* classifier to form classification of 0–9 digits. A test demonstration of LeNet is given in Figure 4.

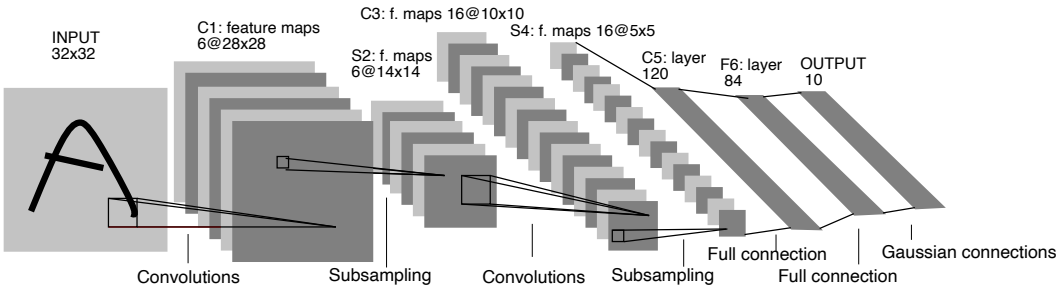


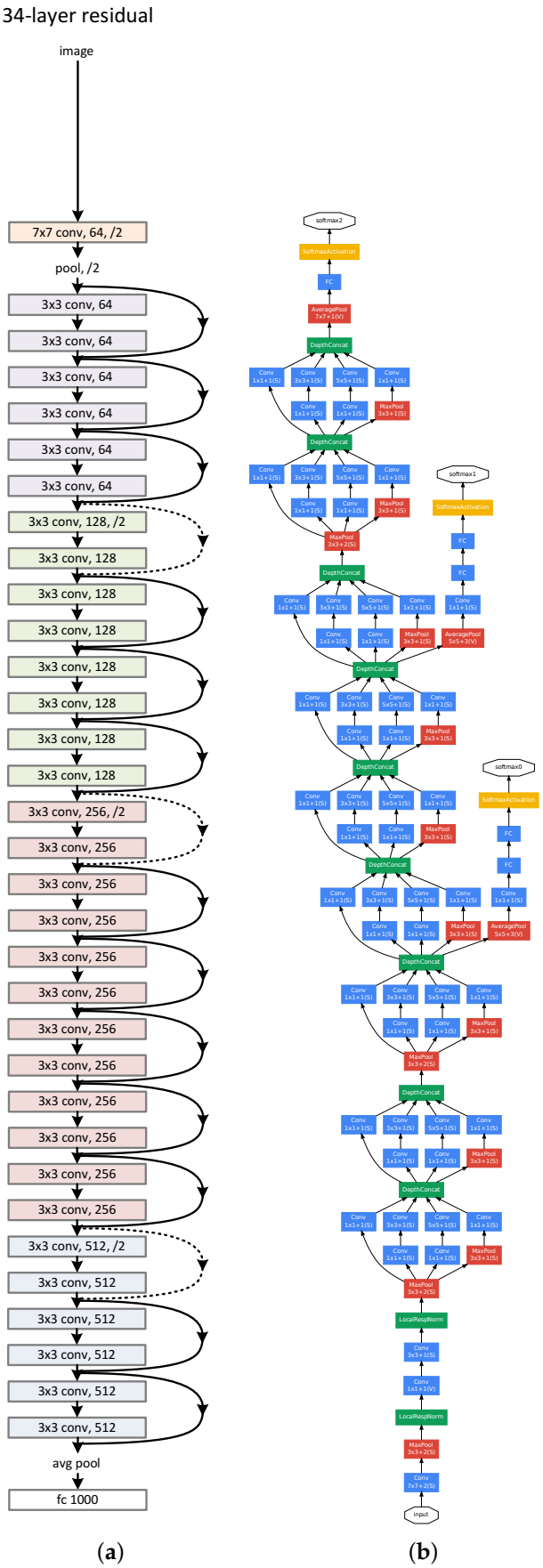
Figure 4. The architecture of LeNet-5 (source: [24]).

An important breakthrough in the domain of computer vision and deep learning is ResNet (also known as Residual Network) [25]. To get better performance in ResNet hundreds of convolution and pooling layers are used. Getting better accuracy in deep neural network is difficult when a large number of layers are used mainly because of the vanishing gradient problem [26]. Vanishing gradient means that when the gradient is computed at later layers of the ResNet and propagated back in the back propagation process, the repeated multiplication of small values of gradients makes it even more small, such that it is value is not significant to make any changes. In ResNet a novel technique known as "Identify Shortcut Connection" is used to deal with vanishing gradient problem. Using this skip connection one or more layers are skipped during the back propagation. A simple demonstration of ResNet with 34 layers is shown on the left side of Figure 5.

Another important milestone in the progress made in computer vision is the development of Inception network [2,27]. The idea of Inception network is based on the observations that the important features in a given image can exist in different variation. Therefore, to perform convolution operation it is not feasible to use an optimal kernel size. Salient features distributed more globally are identified by larger kernels. Those features that are distributed locally are observed by smaller kernels. Different kernel sizes are used at a same level in Inception network. There are nine inception modules that are stacked linearly. There are 27 layers (including pooling layers) and average pooling is used. Auxiliary classifier are used to handle vanishing gradient problem in Inception network. A sample demonstration of the Inception network is given at the right hand side of Figure 5.

3.3.2. Tree-CNN

We implemented and tested most popular CNN models such as ResNet and Inception model on the Wedding dataset and as demonstrated in Section 4 we are not able to reduce the gap of geo-diversity in open datasets. A model trained on ImageNet is not able to learn and increase the knowledge space from Wedding dataset, and vice versa i.e., the models trained on Wedding dataset is not able to share information with models trained on ImageNet. In this section, we devised a technique based on Tree-CNN [28], where models are able to share information gained from different datasets and hence we are able to reduce the gap of geo-diversity in open datasets.



(**a**) (**b**)

Figure 5. (**a**) ResNet architecture (Source: [25]). (**b**) Inception architecture (Source: [2]).

In the Tree-CNN model, different nodes are connected in the form of a tree. Every node has a CNN and is trained to make classification on the input into children nodes. The leaf nodes are the last step of classification. The intermediary nodes make a classification to coarse classes and leaf nodes make classification to exact classes. We took the ResNet model that is initially trained on a large image dataset known as ImageNet and is able to recognize 1000 classes. In Tree-CNN it is represented as a one root node and 1000 leaf nodes. We want to transfer the learning from ImageNet to Wedding datasets. In the Wedding datasets, we have eight classes; six classes belong to *Person* class and two classes belong to *Car* class. In ImageNet *Person* class has three different types i.e., *scuba diver*, *bridegroom* and *baseball player*. The *car* class has types *Race car*, *car mirror*, *passenger car*, *beach wagon*, *freight car* and *bumper car*.

The Tree-CNN will learn to make classification of the eight classes in the Wedding dataset while maintaining the knowledge of ImageNet. We start from the root node and provide the Wedding dataset. The output layer produce a 3-dimensional matrix represented by $O^{k\times M\times I}$, where K represent the number of children of root node, the number of newly introduced classes i.e., 8 is represented by M and the number of images in a given class is represented by I. $O(k,m,i)$ shows the output from the kth neuron of the ith image that belongs to the mth class where $K \in [1,K]$, $m \in [1,M]$, and $i \in [1,I]$. The values of outputs are averaged over I images and represented by $O_{avg}^{K\times M}$ and the *softmax* is computed as shown in Equation (1). The result of the $softmax$ is the probability matrix $L^{K\times M}$ as shown in Equation (2). The probability matrix contains the probability of creating new node or merging two nodes.

$$O_{avg}^{(k,m)} \leftarrow \sum_{i=1}^{I} \frac{O(k,m,i)}{I} \tag{1}$$

$$L(k,m) \leftarrow \frac{e^{O_{avg}^{(k,m)}}}{\sum_{k=1}^{K} e^{O_{avg}^{(k,m)}}} \tag{2}$$

An ordered list S is created from $L^{K\times M}$, which have the following characteristics:

- The list consists of M objects and corresponds to M new classes.
- Every object S[i] has the following features:
  - The label of new class is stored in S[i].label
  - The top 3 average *softmax* (O_{avg}) output values are stored in S[i].values as a vector [v1, v2, v3] where $v1 \geq v2 \geq v3$
  - The nodes corresponding to the softmax values $v1, v2, v3$ are stored in S[i].nodes.
- S is ordered in the descending order of S[i].value[1]

The sorting of S ensures that newly created classes with large probability are combined with the Tree-CNN. After S is constructed, we analyse the first element i.e., S[1] and take one of the 3 actions:

1. Addition of newly created class to already present node: If v^1 is larger than v^2 by the threshold α, it shows a high correlation with that child node. Therefore, the newly created class is combined with the child node n_1.
2. Merging children nodes to create a new node and added the newly created class to the node: In case of larger than one children nodes where the newly created class have high probability for, we can combine them to form a new node. This is possible when $v^1 - v^2 < \alpha$ and $v^2 - v^3 > \beta$, where α and β are threshold values provided by user.
3. Add newly created class as a new node: In case the newly created class does not have a probability that is larger than the other values by a threshold ($v^1 - v^2 < \alpha,\ v^2 - v^3 < \beta$) or all children nodes are full, Tree-CNN grows horizontally when new classes are added as a new child node. The node becomes a leaf node to make classification of class.

The values of α and β are defined by user. α denotes the difference of values for nodes n_1 and n_2 i.e., we make sure that both values are not the same and are different by a good margin. If we increase the margin values the difference between the two values increase. Similarly, β denotes the difference of values for nodes n_2 and n_3. Values are top three average *softmax* O_{avg} values for the class in a decreasing order.

An example of how CNN trained on ImageNet is adapted to accommodate new classes from the Wedding dataset is given in Figure 6. In the Wedding data set, there are labels that belong to *Persons* class of ImageNet. For example *Not Groom*, *Formal*, *Not Formal* are labels of *Persons* class. However, then there are two labels *Bride Not Bride* that do not make a group with other labels of the *Persons* class and hence two new branches are created as *Male* and *Female* and the new classes i.e., *Bride*, *Not Bride* are placed in the *Female* branch.

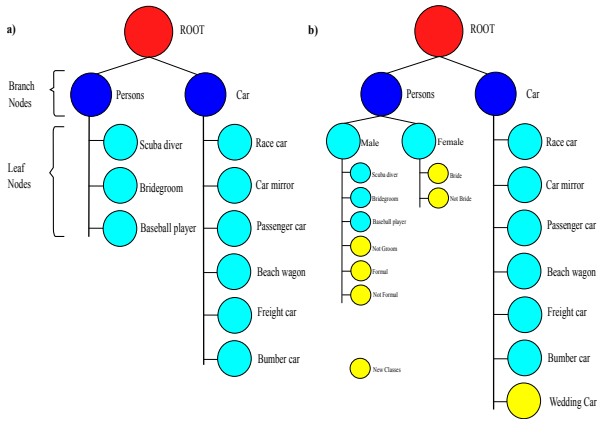


Figure 6. An example demonstration of how Tree-CNN learns new classes from the Wedding data set and adds new classes along with the classes already learned in ImageNet. (**a**) Shows the Tree-CNN on ImageNet (**b**) Shows how the Tree-CNN learns new features from Wedding data set and adds with the labels already learned from ImageNet data set.

Algorithm 1 describes the pseudo-code of Tree-CNN.

Algorithm 1: Algorithm of Tree-CNN

```
S = CreateS(L, Node, maxChildren)  // L is Matrix of probabilities, Node is the current node
    in the tree, maxChildren represents the maximum number of children per node.  S is an ordered
    list from L
while there is an element in S do
    [label, value, node] = GetFeatures(S[1]) // Get features of the first object
    if value[1] - value[2] > α then
        // The new class has a strong preference for node[1]
        Root = Class_Addition_to_Node(Root, label, node[1])
    else
        if value[2] - value[3] > β then
            // The new class has similar strong preference for node[1] and node[2]
            Root = Combine_node(Root, node[1], node[2]) // combine node[2] into node[1]
            Root = Class_Addition_to_Node(Root, label, node[1])
        else
            Small_node = Nodes_with_smaller_Children(node[1], node[2])
            // Add new class to the smaller output node
            Root = Class_Addition_to_Node(Root, label, node[1])
```

4. Simulation

4.1. Configuration of Machine

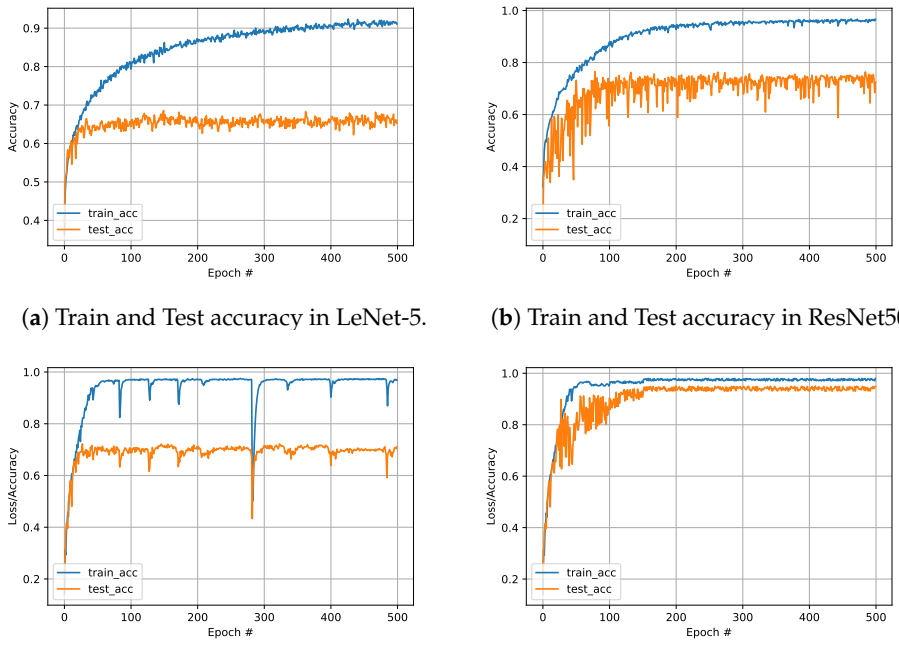
All experiments in this paper are executed using *nvidia GeForce GTX 1080*. It consists of 2560 CUDA cores, with the Boost Clock of 1733 MHz and the size of the memory is 8 BG GDDR5X.

4.2. CNN Architectures Trained on Wedding Dataset

We executed the four diverse CNN models (LeNet, ResNet, Inception and Tree-CNN) for 500 epochs, using Adam [29] optimization technique using Keras library function and having a 0.001 learning rate. Weights and bias are initialized with Glorot Uniform initializer [30]. Random search technique on the grid is used to fine-tune different parameters such as diverse combination of layers, units per layer, diverse optimization techniques. The training is performed using TensorFlow [31] and have exploited parallelism of CUDA cores in GPU.

4.3. Accuracy of CNN Models in Train and Test Dataset

The performance in train and test datasets in different iterations for all models is shown in Figure 7. The accuracy in train set on LeNet is shown in Figure 7a which increases per each iteration and crosses 90%. The accuracy on test set reaches to the 65%. The accuracy in test set is not increasing, but there are a few classes that the model trained on the train set gets over-fitting. The accuracy on test dataset increases up to 76% in ResNet50 as shown in Figure 7b. Similarly the accuracy on test data set is also reached up to 76% in Inception as shown in Figure 7c. As it is observed that ResNet50 and Inception perform better than LeNet-5, demonstrating that as we increase the quantity of layers and increase the size of the dataset, the accuracy tends to improve. There is vanishing gradient problem [26], but ResNet and Inception architectures have developed different mechanisms of dealing with the problem as explained in [2,25,27]. The accuracy in Tree-CNN is shown in Figure 7d and the test accuracy goes above 90% as it shares the features learned from ImageNet and Wedding dataset.



(**a**) Train and Test accuracy in LeNet-5.

(**b**) Train and Test accuracy in ResNet50.

(**c**) Train and Test accuracy in Inception.

(**d**) Train and Test accuracy in Tree-CNN.

Figure 7. Train and Test accuracy by LeNet, ResNet, Inception and Tree-CNN in classification of Wedding dataset.

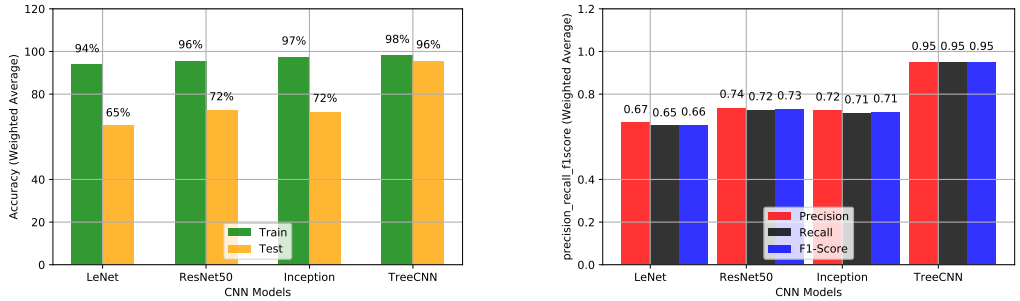
$$\begin{aligned} Accuracy &\leftarrow \frac{TN + TP}{TN + TP + FN + FP} \\ Precision &\leftarrow \frac{TP}{TP + FP} \\ Recall &\leftarrow \frac{TP}{TP + FN} \\ F1 - Score &\leftarrow 2 \times \frac{Precision \times Recall}{Precision + Recall} \end{aligned} \tag{3}$$

4.4. Performance in Terms of Accuracy, Precision, Recall and F1-Score

Accuracy, Precision, Recall and F1-Score are computed using formulae shown in Equation (3). We have represented True Positive by TP, True Negative represented by TN, False Positive represented by FP and False Negative represented by FN. For different CNN models i.e., LeNet, ResNet, Inception and Tree-CNN we computed accuracy, precision, recall and F1-score and have shown in Figure 8. The performance in terms of accuracy of the models is shown in Figure 8a. It is observed that ResNet and Inception have better test accuracy as compared to LeNet. However, Tree-CNN achieves a test accuracy up to 96%, and demonstrates how sharing knowledge of ImageNet helps the model to gain knowledge from Wedding dataset and adds to the knowledge base. Similarly precision, recall and F1-score achieved by ResNet and Inception are improved compared to LeNet as shown in Figure 8b. However, the precision, recall and F1-score in Tree-CNN is 0.95 and demonstrates higher accuracy in Tree-CNN as it shares knowledge space.

4.5. Confusion Matrix

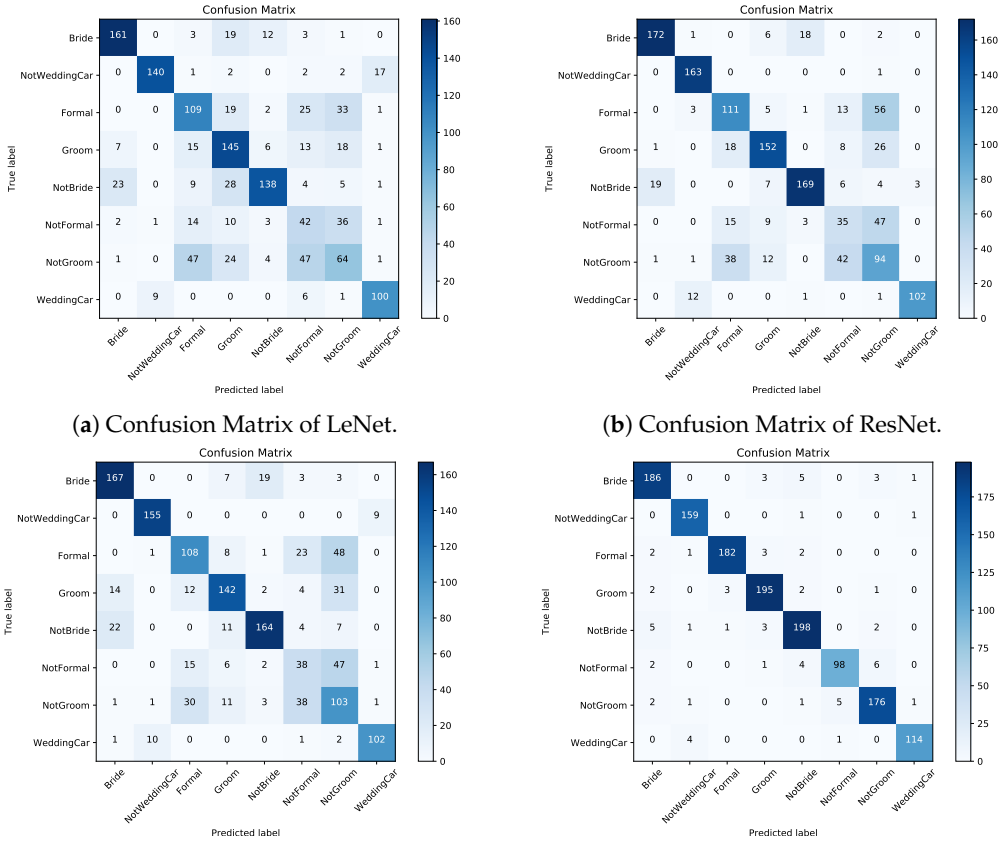
Another performance estimation procedure in machine learning classification issues is known as Confusion Matrix. In binary classification, the size of the table is 2×2 showing true positive, true negative, false positive and false negatives. However, in multi-class classifications problems the size of the table rise to the square of the number of classes. For classification models LeNet, ResNet, Inception and Tree-CNN the confusion matrix in obtaining prediction of classes in the Wedding dataset is shown in Figure 9. The performance of the model is high when diagonal of the confusion matrix has large values. It can be observed in the Figure 9, for Inception and ResNet the values are large on diagonals, demonstrating that these models are more accurate than LeNet. The confusion matrix for Tree-CNN is even more better than ResNet and Inception, demonstrating that training from scratch does not mean improvement in performance, but sharing knowledge helps us to improve performance while at the same time reduces the gap of geo-diversity in open datasets.



(**a**) Train and test accuracy in LeNet, ResNet, Inception and Tree-CNN.

(**b**) Precision, Recall and F1-Score computed by LeNet, ResNet, Inception and Tree-CNN.

Figure 8. Accuracy, Precision, Recall and F1-Score computed by LeNet, ResNet, Inception and Tree-CNN in making predictions in Wedding dataset. Figure (**a**) shows the train and test accuracy in CNNs and Figure (**b**) shows the Precision, Recall, F1-Score by different CNNs.



(**a**) Confusion Matrix of LeNet.

(**b**) Confusion Matrix of ResNet.

(**c**) Confusion Matrix computed by Inception.

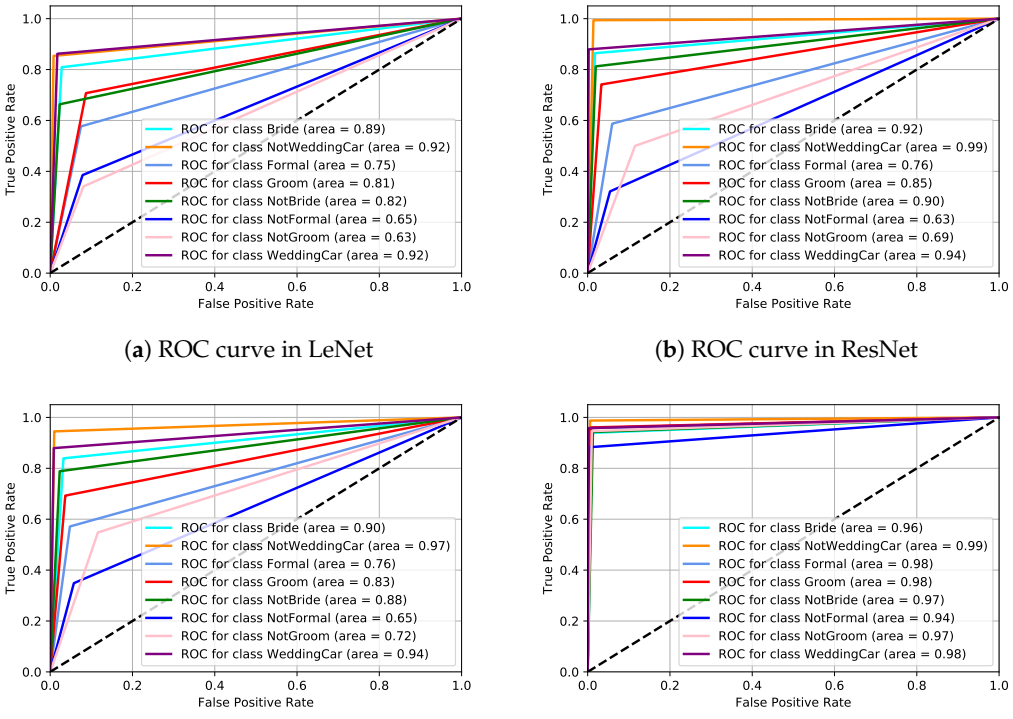
(**d**) Confusion Matrix computed by Tree-CNN.

Figure 9. Confusion Matrix computed by LeNet, ResNet, Inception and Tree-CNN in the prediction of Wedding dataset.

4.6. Evaluation with ROC Curve

The performance in classification by CNN models is demonstrated by using ROC (Receiver Operating Characteristic) curve. True Positive Rate (TPR) and False Positive Rate (FPR) are focused on the ROC curve. The calculation of these parameters is given in Equation (4). The ROC curve for CNN models i.e., LeNet, ResNet, Inception and Tree-CNN is given in Figure 10. The accuracy of CNN models is checked in ROC curve by covering the area of lines on the left side of the diagonal. The larger the area the higher the accuracy. As it can be observed in the figures, the lines are more on the left side of the diagonal for ResNet and Inception model, demonstrating the better accuracy of these models compared to LeNet. However, the ROC for Tree-CNN shown in Figure 10d is more on the left side and making area under the curve from 0.94 to 0.99. This demonstrates that Tree-CNN is the most suitable model when you want to learn new features but at the same time retain the previous knowledge.

$$\begin{aligned} TPR &\leftarrow \frac{TP}{TP+FN} \\ FPR &\leftarrow \frac{FP}{FP+TN} \end{aligned} \tag{4}$$



(**a**) ROC curve in LeNet

(**b**) ROC curve in ResNet

(**c**) ROC curve in Inception

(**d**) ROC curve in Tree-CNN

Figure 10. ROC curve obtained by LeNet, ResNet, Inception and Tree-CNN computed by making prediction in Wedding dataset.

4.7. Prediction of Labels by Different Models

The classification of images taken from Wedding dataset and classified using CNN models pre-trained on ImageNet and CNN trained from scratch are shown in Table 2. The prediction of labels using ResNet and Inception pre-trained on ImageNet are shown in the second and third columns. It can be observed that most of the labels are not appropriate or disturbing. For example, for the image in the first row a bride is labelled as abaya, vestment, cloak and theater curtain which are not appropriate. In the image in the third row a man dressed in a regional formal dressed is labelled as bulletproof vest, windsor tie, gar or barracouto, which are disturbing. In the image in the fourth row, labels are predicted as military uniform, pickelhoube, fur coat, bow tie etc, which are not appropriate to be used for a South Asian groom. In the image in fifth row, labels are predicted as groom, feather bos, fountain, stole, sarang etc are used to describe a typical South Asian woman. In the image in the sixth row a typical South Asian dressed man is labelled as file, refrigerator, photocopier, Loafer etc, which are disturbing. In the image in the seventh row a formal dressed man is labelled as bow tie, microphone, mask, jersey, drumstick which are not appropriate. These inappropriate labelling is happening because the CNN models are pre-trained on ImageNet which contains images mostly from Europe, America and Australia. When images from South Asian regions are given to the model, the CNN model predicts inappropriate labels.

We trained LeNet, ResNet and Inception CNN models from scratch on the wedding dataset and the results are shown in the fourth, fifth and sixth columns respectively. These models achieved better accuracy than ResNet and Inception pre-trained models, but the problem is that these models trained from scratch suffer from catastrophic forgetting i.e., these models learn new labels, but do not keep the labels that are learned in ImageNet. To deal with this problem, we have taken TreeCNN pre-trained on ImageNet and then trained on Wedding dataset. This way it not only learns new labels in the Wedding data, but at the same time retains the knowledge gained from ImageNet.

Table 2. The classification of images from the Wedding dataset in different CNN models trained from scratch or pre-trained using ImageNet.

Image	ResNet50-Pre-Trained	Inception-V3-Pre-Trained	LeNet-Scratch	ResNet-Scratch	Inception-Scratch	Tree-CNN
	abaya: 30.1%, vestment: 23.2%, cloak: 5.0%, theater_curtain: 5.0%	abaya: 31.6% harp: 18.9% vestment: 8.4% wig: 3.1%	Bride: 100.00% Groom: 0.0%	Bride: 100.00% Groom: 0.00%	Bride: 87.04% NotBride: 12.94%	Bride: 90.2%, wig: 7.4%, NotBride: 2.2%, bridegroom: 0.1%
	beach_wagon: 41.7%, pickup:10.9%, car_wheel: 8.7%, cab: 6.8%	jeep: 54.4%, beach_wagon: 25.9%, pickup: 5.3%, car_wheel: 1.7%	NotWeddingCar: 72.89% NotFormal: 26.23%	NotWeddingCar: 99.99% WeddingCar: 0.01%	NotGroom: 99.89% NotWeddingCar: 0.08%	minivan: 96.8%, beach_wagon: 1.8%, moving_van: 1.3%, parking_meter: 0.1%
	buletproof_vest: 43.6%, windsor_tie: 6.9%, gar: 2.7%, barracouto: 2.6%	bulletproof_vest: 33.1%, Windsor_tie: 5.2%, paddle: 2.1%, barracouta: 2.0%	Formal: 63.56% NotFormal: 22.51%	Formal: 94.94% NotGroom: 5.04%	NotGroom: 93.88% Formal: 6.02%	Formal: 98.1%, Groom: 1.6%, cardigan: 0.2%, suit: 0.0%
	fur_coat: 21.0%, breastplate: 8.8%, bow_tie: 7.5%, cardigan: 6.9%	military_uniform: 7.9%, pickelhoube: 7.2%, fur_coat: 6.6%, bow_tie: 4.7%	Groom: 100.00% NotGroom: 0.0%	Groom: 100.00% NotGroom: 0.00%	Groom: 100.00% NotGroom: 0.00%	bridegroom: 99.2%, NotGroom: 0.4%, NotFormal: 0.3%, mask: 0.1%
	groom: 16.0%, feather_boa: 14.8%, fountain: 4.4%, stole: 4.1%	sarang: 36.1%, maillat: 6.1%, gown: 4.7%, maillot: 4.3%	NotBride: 100.00% Bride: 0.0%	NotBride: 100.00% Bride: 0.00%	NotBride: 99.94% Bride: 0.06%	NotBride: 96.1%, Bride: 0.2%, cloak: 2.7%, Sarang: 1.0%
	file: 18.3%, refrigerator: 8.6%, photocopier: 3.5%, desk: 3.3%	suit: 67.1%, Loafer: 5.8%, Windsor_tie: 1.7%, sweatshirt: 1.2%	NotFormal: 99.16% NotGroom: 0.40%	NotFormal: 92.08% Formal: 6.60%	NotGroom: 100.00% Groom: 0.00%	NotFormal: 90.8%, Loafer: 4.3%, Formal: 1.5%, jean: 3.4%
	bow_tie: 30.1%, Windsor_tie: 5.4%, microphone: 4.5%, mask: 4.5%	drumstick: 6.5%, jersey: 4.2%, sweatshirt: 3.3%, mask: 2.8%	NotGroom: 99.18% NotFormal: 0.82%	NotGroom: 100.00% Groom: 0.00%	NotGroom: 100.00% NotFormal: 0.00%	NotGroom: 98.2%, jean: 1.7%, NotFormal: 0.0%, suit: 0.0%
	gondola: 21.8%, clog: 19.9%, minivan: 10.7%, milk_can: 6.5%	pickelhaube: 6.9%, waffle_iron: 5.9%, minivan: 5.5%, space_bar: 4.2%	WeddingCar: 100.00% NotWeddingCar: 0.0%	WeddingCar: 100.00% Groom: 0.00%	WeddingCar: 99.98% NotGroom: 0.01%	WeddingCar: 94.2%, altar: 4.2%, limousine: 1.6%, pot: 0.0%

In Figure 2 it is shown that *bridegroom* from Europe/US/Australia is easily classified, while the model had low accuracy to classify *bridegroom* from Pakistan, India, Ethiopia etc. In our model trained on Tree-CNN, we are able to classify brides and bridegroom from South Asian countries, similar to the classification of brides and bridegroom from the regions having high representation in ImageNet dataset. This classification is shown in Figure 11.

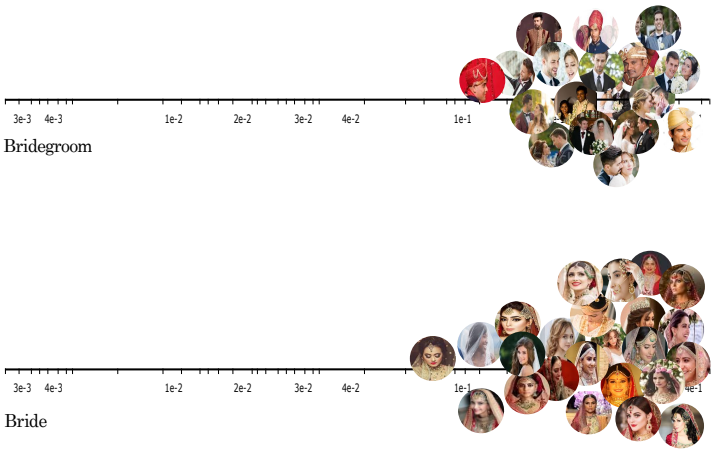


Figure 11. Images of *bridegrooms* and *bride* taken from different regions and represented as a log-likelihood using model pre-trained with ImageNet and then learn new classes from the Wedding dataset.

4.8. Discussion

We developed a model based on Tree-CNN that is able to extend its knowledge that is learned from ImageNet to another dataset Wedding to learn new classes. For example, bridegroom is a class in ImageNet, but Wedding data set has classes "Bride", "Not Bride", "Groom", "Not Groom", etc. The model based on Tree-CNN is able to learn these new classes while keeping the knowledge learned from ImageNet dataset. If the model is further trained with a new dataset it will be able to extend its knowledge and learn the new classes in the new dataset. This way it is able to reduce the gap of geo-diversity for developing world. Which means, that although there is no or little representation for developing world in the ImageNet or any other standard image dataset, the proposed model can help to learn new classes from a newly developed dataset while keeping the existing knowledge.

Our results have demonstrated that the model proposed in this paper which is based on Tree-CNN is able to learn new classes, which was not able to better captured by popular CNN models trained from scratch. The results of pre-trained models on ImageNet dataset are also shown and have demonstrated that the accuracy is very low. The proposed model based on Tree-CNN has demonstrated an accuracy of 98%. This demonstrates that the model has learned new classes from the newly developed dataset and has extended its knowledge.

5. Conclusions

Computer vision problems are solved by using CNN trained on different publicly available datasets such as ImageNet or Open Images. However, most of the public datasets contain images from economically developed countries such as Europe, America and Australia and the developing worlds have not representation in the open datasets. The paper studies the problem of no classification without enough representation in detail, analyzing existing public datasets and the classification of newly created datasets based on popular CNN models pre-trained on open datasets. The paper demonstrates that the results of popular models which are trained from scratch on the dataset developed in this paper to demonstrate how classification accuracy is improved when there is representations of different regions. However, the training from scratch on new datasets creates another problem of catastrophic forgetting i.e., the model is not able to retain the knowledge it has gained from the previous dataset. The paper also use pre-trained models on open datasets and make classification of different images in Wedding dataset and demonstrate that the performance is not good and the labels are disturbing sometimes because of no representation from the developing world. This paper presents a novel technique of hierarchical CNN, where the knowledge gained from Open datasets are retained, and is able to learn new features from the new datasets. We demonstrated that Tree-CNN model is able to improve accuracy by more than 95% as compared to training currently popular models from scratch.

As a future direction, we will further investigate the effect of more learning to increase its knowledge of the already learned classes. For instance, in this manuscript we demonstrated that the knowledge is increased to learn new classes in the Wedding dataset, but whether it has any effect on the previously learned classes such as dog species or cars etc is not explored. This is one possible future direction of this research to investigate it further.

Author Contributions: Conceptualization, H.A., A.A. and I.U.; Formal analysis, H.A., A.A. and I.U.; Investigation, I.U.; Methodology, H.A.; Project administration, A.A.; Validation, I.U.; Writing—original draft, H.A.; Writing—review & editing, A.A. and I.U. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Taif University Researchers Supporting Project number (TURSP-2020/231), Taif University, Taif, Saudi Arabia.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Shankar, S.; Halpern, Y.; Breck, E.; Atwood, J.; Wilson, J.; Sculley, D. No Classification without Representation: Assessing Geodiversity Issues in Open Data Sets for the Developing World. *arXiv* **2017**, arXiv:1711.08536.
2. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9. [CrossRef]
3. Lawrence, S.; Giles, C.; Tsoi, A.; Back, A. Face recognition: A convolutional neural-network approach. *IEEE Trans. Neural Netw. Learn. Syst.* **1997**, *8*, 98–113. [CrossRef] [PubMed]
4. Bobić, V.N.; Tadić, P.R.; Kvascev, G. Hand gesture recognition using neural network based techniques. In Proceedings of the 2016 13th Symposium on Neural Networks and Applications (NEUREL), Belgrade, Serbia, 22–24 November 2016; pp. 1–4.
5. Gu, J.; Wang, G.; Cai, J.; Chen, T. An Empirical Study of Language CNN for Image Captioning. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 1231–1240. [CrossRef]
6. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
7. Awasthi, A.; Sarawagi, S. Continual Learning with Neural Networks: A Review. In Proceedings of the ACM India Joint International Conference on Data Science and Management of Data, Kolkata, India, 3–5 January 2019; pp. 362–365. [CrossRef]
8. Silver, D.L.; Yang, Q.; Li, L. *Lifelong Machine Learning Systems: Beyond Learning Algorithms*; 2013 AAAI Spring Symposium Series; AAAI: Stanford, CA, USA, 2013; Volume SS-13-05.
9. Sultana, F.; Sufian, A.; Dutta, P. Advancements in Image Classification using Convolutional Neural Network. In Proceedings of the 2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN), Copenhagen, Denmark, 19 August 2018; pp. 122–129.
10. Boulent, J.; Foucher, S.; Théau, J.; St-Charles, P.L. Convolutional Neural Networks for the Automatic Identification of Plant Diseases. *Front. Plant Sci.* **2019**, *10*, 941. [CrossRef] [PubMed]
11. Jmour, N.; Zayen, S.; Abdelkrim, A. Convolutional neural networks for image classification. In Proceedings of the 2018 International Conference on Advanced Systems and Electric Technologies, Hammamet, Tunisia, 22–25 March 2018; pp. 397–402. [CrossRef]
12. Sharma, N.; Jain, V.; Mishra, A. An Analysis Of Convolutional Neural Networks For Image Classification. *Procedia Comput. Sci.* **2018**, *132*, 377–384. [CrossRef]
13. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
14. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]

15. He, T.; Zhang, Z.; Zhang, H.; Zhang, Z.; Xie, J.; Li, M. Bag of Tricks for Image Classification with Convolutional Neural Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
16. Lee, S.; Stokes, J.; Eaton, E. Learning Shared Knowledge for Deep Lifelong Learning using Deconvolutional Networks. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, Macao, China, 10–16 August 2019; pp. 2837–2844. [CrossRef]
17. Yang, K.; Qinami, K.; Li, F.-F.; Deng, J.; Russakovsky, O. Towards Fairer Datasets: Filtering and Balancing the Distribution of the People Subtree in the ImageNet Hierarchy. In Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency (FAT*'20), Barcelona, Spain, 27–30 January 2020; Association for Computing Machinery: New York, NY, USA, 2020; pp. 547–558.
18. Jia, Y.; Batra, N.; Wang, H.; Whitehouse, K. A Tree-Structured Neural Network Model for Household Energy Breakdown. In Proceedings of the World Wide Web Conference, New York, NY, USA, 17 May 2019; pp. 2872–2878. [CrossRef]
19. Jiang, S.; Xu, T.; Guo, J.; Zhang, J. Tree-CNN: From generalization to specialization. *EURASIP J. Wirel. Commun. Netw.* **2018**, *2018*, 216. [CrossRef]
20. Ni, J.; Gong, T.; Gu, Y.; Zhu, J.; Fan, X. An Improved Deep Residual Network-Based Semantic Simultaneous Localization and Mapping Method for Monocular Vision Robot. *Comput. Intell. Neurosci.* **2020**, *2020*, 1–14. [CrossRef] [PubMed]
21. Daliri, S. Using Harmony Search Algorithm in Neural Networks to Improve Fraud Detection in Banking System. *Comput. Intell. Neurosci.* **2020**, *2020*, 1–5. [CrossRef] [PubMed]
22. Chen, G.; Lu, G.; Xie, Z.; Shang, W. Anomaly Detection in EEG Signals: A Case Study on Similarity Measure. *Comput. Intell. Neurosci.* **2020**, *2020*, 1–16. [CrossRef] [PubMed]
23. Wang, H.; Zhang, Y.; Yu, X. An Overview of Image Caption Generation Methods. *Comput. Intell. Neurosci.* **2020**, *2020*, 1–13. [CrossRef] [PubMed]
24. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
25. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. *arXiv* **2015**, arXiv:1512.03385.
26. Bengio, Y.; Simard, P.; Frasconi, P. Learning Long-term Dependencies with Gradient Descent is Difficult. *Trans. Neur. Netw.* **1994**, *5*, 157–166. [CrossRef] [PubMed]
27. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
28. Roy, D.; Panda, P.; Roy, K. Tree-CNN: A hierarchical Deep Convolutional Neural Network for incremental learning. *Neural Netw.* **2020**, *121*, 148–160. [CrossRef] [PubMed]
29. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization, 2014. In Proceedings of the 3rd International Conference for Learning Representations, San Diego, CA, USA, 7–9 May 2015.
30. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS'10), Sardinia, Italy, 13–15 May 2010.
31. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. TensorFlow: A System for Large-scale Machine Learning. In Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation, Berkeley, CA, USA, 4 November 2016; pp. 265–283.