ISSN: 2633-4828 Vol. 2 No.1, June, 2020

International Journal of Applied Engineering & Technology

TRANSFER LEARNING BASED FINE-GRAINEDBIRDS IMAGE CLASSIFICATION

Tahreem. Hamid[1] and Waqar. Ahmad[2]

[1]University of Engineering and Technology Taxila, Pakistan

[2]Department of Computer Engineering, University of Engineering and Technology Taxila, Pakistan

[1]tahreem.hamid@students.uettaxila.edu.pk and [2]waqar.ahmad@uettaxila.edu.pk

Received: 16th March 2020 Revised: 30th May 2020 Accepted: 19th June 2020

ABSTRACT

For categorizing different objects in their respective class, image classification is one of the most hot topic in research domain. One of the most interesting topic is the birds classification which has great importance due to the large intra and small inter class variations. To classify sub-classes under one main class is the most challenging task. NasNet Large is a convolutional neural network (CNN) architecture that was trained on a large dataset of images called ImageNet. In this study, the researchers used NasNet Large as a pre-trained model for transfer learning to classify images of birds. We fine-tuned the pre-trained model

on a dataset of bird images and evaluated its performance on a test set. The results showed that the fine-tuned NasNet Large model achieved a high accuracy in classifying bird images, demonstrating the effectiveness of transfer learning for this task. Initially perform some pre- processing techniques on the selected dataset and then extract features by using a novel approach that combines the in-depth information to create more discriminatory features by using a pre-trained model. Furthermore, a robust feature selection and dimension reduction method called Entropy-Controlled Neighborhood Component Analysis (ECNCA) was introduced in the second stage. This technique aimed to identify the most informative and relevant features from the extracted set of features. By reducing the dimensionality of the feature space, ECNCA helps in reducing noise, enhancing classification performance, and improving computational efficiency. The results of the study indicated that the proposed methodology, incorporating the pre-trained NasNet Large model, feature extraction, and ECNCA, achieved a high classification accuracy of 92.6% on the CUB-200-2011 dataset. This suggests that the combination of transfer learning, feature extraction, and dimension reduction techniques can effectively classify birds with discriminating features.

INDEX TERMS Convolution Neural Networks (CNNs), Caltech-UCSD Birds-200-2011 (CUB- 200-2011), Entropy-Controlled Neighborhood Component Analysis (ECNCA)

I. INTRODUCTION

In recent years, researchers made tremendous role in an image classification. Image classification research plays a vital role in many areas like in distinguishing difference between categories and their sub-categories. It used to lesson the distance between computer and human and can also use to train large amount of supervised and unsupervised data You are correct that image classification research has made significant advancements in recent years. It plays a crucial role in various domains, including distinguishing differences between categories and sub-categories of objects in images. Image classification helps bridge the gap between computers and humans by enabling machines to understand and interpret visual information.

One of the key benefits of image classification is its ability to handle large amounts of data, both in supervised and unsupervised learning scenarios. Supervised image classification involves training models on labeled datasets, where each image is associated with a specific category or class. This allows the model to learn patterns and features that differentiate different classes, enabling accurate classification.

On the other hand, unsupervised image classification techniques aim to automatically discover patterns and structures within unlabeled datasets. These methods can be useful when labeled data is scarce or when researchers want to uncover hidden relationships or groupings in the data.

Image classification has found applications in various fields, including computer vision, medical imaging, autonomous vehicles, surveillance systems, and many more. It enables tasks such as object recognition, scene understanding, facial recognition, and even fine-grained categorization, where sub-categories within a class need to be accurately identified.

The advancements in image classification research have led to the development of more sophisticated algorithms and deep learning architectures. Convolutional neural networks (CNNs) have emerged as a dominant approach for image classification due to their ability to automatically learn hierarchical features from raw image data. Pre-trained CNN models, such as VGG, ResNet, and Inception, have become widely used for transfer learning, where the knowledge acquired from large-scale datasets like ImageNet can be leveraged to solve new image classification problems.

Overall, image classification research has revolutionized various industries and has tremendous potential for further advancements. It continues to contribute to narrowing the gap between human and computer understanding of visual information and enables a wide range of applications and innovations. [1]. The main concept used for the purpose of image classification is known as machine learning. These machine learning concept based algorithms includes a feature extraction module that extract the features of edges and textures etc [2]. Deep learning introduces a multi-level architecture of different algorithms and expressed as Artifical Neural Network (ANN). ANN model behaves like a human brain and efficient than progressive machine learning algorithms [3]. Neural networks classify images on the bases of their features [4]. This concept of neural networks has been solved by using complete feature extraction model and solved the previous models problems. The feature extraction models extract the most discriminated features from the training images of the dataset.

Different feature extraction methods like GIST [5], Local Binary Patterns (LBP) [6], Histogram of Oriented Gradient (HoG) [7] and Scale Invariant Feature Transform (SIFT) [8] used to get key features for the purpose of object detection and classification [9]. The study of feature extraction [10] is to automate feature extraction and learning methods. Deep CNNs have both generic [11–13] and fine-grained image classifiers [14, 15] by using these models on ImagNet dataset to classify 1000 different categories [16]. Recently, Convolutional Neural Networks (CNNs) introduced [17–21] by using deep features image classification leading toward improved performance. Deep neural network served as end to end classifier for the purpose of segmentation [22] and detection [23]. To get rid of the over-fitting problem here is the most common solution which is transfer learning [14, 24]. Recent work on this concept [24] show improved accuracy by using transfer learning.

Fine-grained image classification has gained alot of popularity in recent years [25], unlike the primary classification, in which we have to distinguish basic classes such as birds [26], dogs [27], cars [28]. At the same time, fine-grained image classification aims to discriminate subclasses under the main class [14, 25, 29–31], which is also an challenging task. Birds are known as an excellent index of biodiversity for their ecosystem environment. Contraction in birds population observed worldwide, for the purpose of birds protection it is important to monitor them [32]. Birds classification is a challenging task since there is a significant similarity between sub-categories. It is difficult for a common person to identify the sub-categories only by their physical appearance [32]. Most of the work in this field done by detecting local parts of birds, and numerous researchers studied how to extract features to overcome the difficulties of poses and variations in different views [13, 29, 33–35]. Aside from poses and vision changes, one of the significant challenges in bird classification model is to distinguish between high visual correlation classes [25]. Some methods still face considerable difficulty in the classification of visually similar fine-grained birds classification [14, 19, 36].

An effective fine-grained image classification system required that will be helpful in many practical applications.

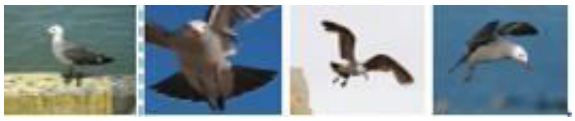
A. PROBLEM STATEMENT

Birds image classification is a most difficult task due to the large dispersion within the class and there also exist a small dispersion between different classes. As there are more than 100 sub-categories under a basic category [37] as shown below in Figure 1, which is difficult to classify between them. The previous methods for the birds image classifications had cost and time consumption as well as accuracy trade off. Researchers suggest that transfer learning is an effective way to increase the accuracy of a deep network [38] and as well as it will consume lesser time in training. So there still exist a gap and we want to explore want to explore few of them which includes:

1) How to solve the problem of large and small class difference and also within the class differences ?
2) How can we get better performance on a fine-grained dataset in a limited amount of time and cost?
3) How to conduct transfer learning on a fine-grained dataset which produces good results?

(a)California Gull

(b)Glaucous Gull



(C) Heermann Gull

(d) Slaty-backed Gull

Figure 1: Four different species from CUB-200-2011 demonstrate the problem of image classification: large within the class variance and small variance between class.

This paper utilizes an experimental approach to analyze concepts of transfer learning. We have to perform aset of experiments to evaluate the proposed methodology.

B. CONTRIBUTIONS

The main contributions of this presented work summarized as follows:

1) A transfer learning approach used for bird's image classification and take advantage of the behavior of selected layers of the Deep Architecture (NASNet Large) on classifier's performance.
2) According to our studies, we have improved the deep model's accuracy with a small learning rate and maintaining the weight of the initial frozen layer to avoid distortion.

3) A hierarchical framework for characteristic selection and dimension reduction proposed. It is initially based on entropy for property selection and uses the following to reduce dimensions: Near Component Analysis (NCA). The corresponds to an average reduction of 92.6% and displays improved classification results.

4) To our knowledge, most of the existing literature does not degrade the profound properties at this level in this area.

C. PAPER ORGANIZATION

The rest of the article was presented as follows: The related work described in Section II. The methodology proposed in Section III. In this Section IV, complete experiments are conducted and an analysis of the presented simulation. Finally, the results concluded in Section V.

II. RELATED WORK

A. *Fine-Grained Image Classification*

Most of the researchers monitored bird species for various purposes, such as ecosystem changes and population investigation [36]. Deep learning models have shown greater performance as compared to machine learning. Deep learning models performed well by using large amount of dataset [39]. Convolutional Neural Network (CNNs) is mostly used for the purpose of object detection and recognition by using the standard ImageNet dataset [11]. Most of the researchers used the concept of discriminant features to achieve more accuracy and followed the concept of pipeline [40]. The concept of bilinear pooling used for second order bilinear feature extraction for birds' classification [41]. To achieve more accuracy by using subtle differences and visual attention [42–45] and to get more information about birds including their parts by using pixel values [14, 36, 46], attributes [47, 48], human interactions [49, 50] and text descriptions [51, 52].

To deal with small amount of data in fine-grained classification the concept of data augmentation helped in increasing the original dataset [45, 47, 53, 54]. Data augmentation techniques [55, 56] can be utilized to increase the limited amount of dataset. By applying this concept, the problem of class imbalance can be addressed [57, 58]. Image resolution factor also affects the performance of the model [41, 59]. In the proposed approach we used high resolution of 331x331 images by using the concept of data augmentation to achieve considerable results. To fine-grained birds image classification used a pre-trained model on ImageNet dataset that is NASNet Large deep model [60]. According to our studies, the image resolution factor effects on the accuracy of the model. There are few deep models on birds' images for the purpose of classification to obtain better results from aerial images [61]. We present a simple technique to deal with this problem that gives the best results.

B. *TRANSFER LEARNING*

The concept of transfer learning used to transfer the knowledge of pre-trained models and used directly as a feature extractor for fine-grained image classification [12, 13, 62]. Recently by using the concept of transfer learning many researchers achieved good results [24, 63–65]. Transfer learning concept for image classification used the pre-trained models on ImageNet datasets [65]. Our main difference is that we first selected a more similar dataset for the closest transfer learning. Secondly, deep model as a feature extractor and trained the last layer of the model by using a linear SVM classifier with fully connected layers to train a network with a series of SVM classifiers.

III. PROPOSED METHODOLOGY

This section presents a methodology that helps in achieving an impressive performance on the selected fine-grained CUB-200-2011 dataset. The primary steps focusing on the pre-processing of the dataset in which we have performed two steps images resolution and data augmentation. This section also explained transfer learning concept in which how we have trained fine-grained dataset on the pre-trained NASNet Large deep model as shown below in block diagram Figure 2.

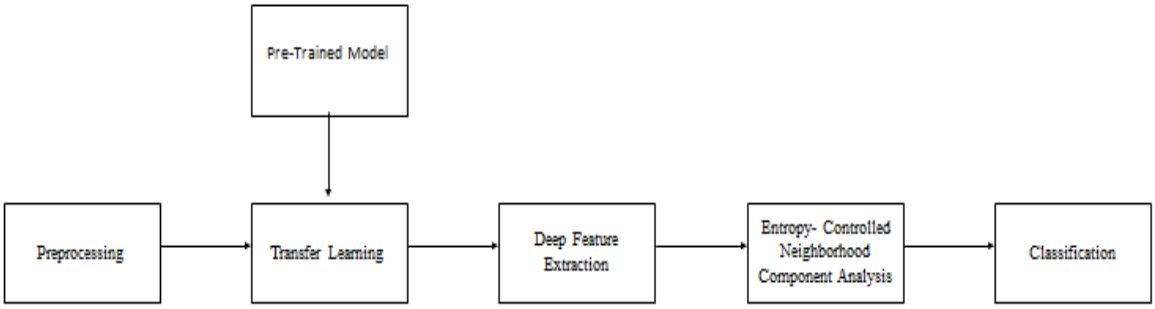


Figure 2: Block Diagram of the Proposed Methodology

A. PRE-PROCESSING

1) Image Resolution

Image pre-processing is the process of preparing images for computer vision tasks, such as image classification, object detection, and segmentation. The goal of pre-processing is to enhance the quality of the images and make them more suitable for the task at hand. There are several common image pre-processing techniques that can beused:

1) Resizing changing the size of the image to a standard size, to make it compatible with the input size of themodel.
2) Normalization converting the pixel values of the image to a range between 0 and 1, or -1 and 1, to makeit compatible with the input range of the model.
3) Grayscale conversion is converting an image from RGB to grayscale, which can be useful for tasks that donot require color information.
4) Histogram equalization is adjusting the image's color balance to improve the contrast and visibility of thefeatures.

In the training phase, the input images are given in square form to the model as an input. As we have seen each model used images of different resolutions such as AlexNet [11] and VGGNet [66]. To stability, the size of the feature map from input image to the fully connected layer remains same. A few advanced deep models like ResNet [67] and Inception [68] used a global average pooling layer followed by convolutional layer. The previous results showed that high resolution images contain more information as well as subtle details which helped with good results. In general, high-resolution images yield better resulnts and as shown in Table 1, from originally 224x224 in AlexNet [11] to 331x331 in recently proposed pre-trained NASNet Large [60]. We also find out that high resolution input images 331x331 in CUB-200-2011 showing impressive performance.

Table 1: Image Resolutions Used by Different Networks

Input Resolution	Networks
224x224	AlexNet [11],VGGNet [66], ResNet [67]
299x299	Inception [68, 69]
320x320	ResNetv2 [70], ResNetXt [71],SENet [72]
331x331	NasNet [60]

2) Data Augmentation

Data augmentation is a technique used to artificially increase the size of a dataset by creating new samples from existing ones. This can be useful in situations where the amount of available training data is limited, as it can help to prevent overfitting and improve the generalization of the model. There are several common data augmentation techniques that can be used:

1) Random cropping, crop a portion of an image and use it as a new sample.
2) Random flipping, horizontally or vertically flipping an image to create a new sample.
3) Random rotation, rotate an image by a random angle to create a new sample.
4) Color jittering, randomly changing the brightness, contrast, and saturation of an image to create a new sample.
5) Gaussian noise, adding Gaussian noise to an image to create a new sample.
6) Random erasing, randomly masking a portion of an image with a random color to create a new sample. Data augmentation should be applied carefully to the dataset, because if it's not done correctly, it might lead to worse performance. Also, different types of data augmentation techniques might be more suitable for different types of data and tasks, therefore, it's important to experiment with different techniques and see which ones.

work best for your specific case.

Deep models utilized a large amount of dataset for training and testing purposes [56]. The even distribution in benchmark datasets such as ImageNet [11] and COCO [58]; image imbalance observed in sub-categories in the CUB-2002011 dataset [26], also observed the poor performance mainly caused by:

1) There is a smaller number of images in the training dataset per class.
2) Class imbalance occurs within each class.

The data augmentation technique applied to increase the number of images in each class by using the zoom, flip and rotate function and as well as to get rid of class imbalance problem.

B. DEEP FEATURE EXTRACTION

Deep features extracted from each layer of a pre-trained model at every stage. The most discriminant features used for exploiting the output layer. Initially the weights kept frozen for the purpose of extracting deep features [73]. A fusion strategy used for utilizing the independent features, which also helps in increasing feature redundancy. Features extracted from each layer of a pre-trained model at every stage. The discriminant features from pre-trained model selected by exploiting the output layer. The initial weights were kept frozen for the purpose of extracting deep features. By utilizing the independent features, a fusion strategy was adopted. Different parts concatenated from pre-trained model to generate a fused feature for the purpose of discriminating features. Fusion strategy increased the feature redundancy.

C. ENTROPY-CONTROLLED NEIGHBORHOOD COMPONENT ANALYSIS

Entropy Controlled Neighborhood Component Analysis (ECNCA) is a technique for non-linear dimensionality reduction and feature extraction [74]. It is based on Neighborhood Component Analysis (NCA), which is a method for preserving the local neighborhood structure of the data in the lower-dimensional space. ECNCA is a variation of NCA that uses entropy as a criterion to find the most informative features [75]. The idea behind it is to find a projection of the data that maximizes the entropy of the class labels while preserving the local neighborhood structure. The ECNCA algorithm consists of the following steps:

1) Construct a k-nearest neighbors' graph of the data.
2) Compute the gradient of the entropy of the class labels with respect to the projection.
3) Optimize the projection by following the gradient and iteratively updating the projection.
4) ECNCA can be useful for feature extraction and dimensionality reduction for image classification, face recognition, and other computer vision tasks. It can be applied as a pre-processing step before training a model to improve its performance.
5) It is worth noting that ECNCA is a complex technique, it has a high computational cost, and it might be impractical for large datasets. Also, it's important to make sure that the extracted features are meaningful and can be used to improve the model's performance.
6) To achieve good results by exploiting a minimum number of features. An Entropy Controlled Neighborhood Component Analysis implemented for the purpose of both feature selection and dimensionalreduction to avoid dimensions issues.

1) Feature Selection

Feature selection is the process of identifying and selecting a subset of relevant features from a large set of features, to improve the performance and interpretability of a model. The goal of feature selection is to reducethe dimensionality of the data and remove irrelevant, redundant, or noisy features.

There are several common feature selection techniques that can be used:

1) Filtering technique evaluating each feature independently based on a predefined criterion and selecting thetop-performing features.
2) Wrapper used for evaluating the performance of a model with different subsets of features and selecting the subset that results in the best performance.
3) Embedded is used for selecting features as a part of the model training process, by considering the featureimportance during the optimization of the model's parameters.
4) Hybrid combining multiple feature selection techniques to improve the performance and robustness of theselection process.

Feature selection should be applied carefully to the dataset, because if it is not done correctly, it might lead to worse performance. Also, different types of feature selection techniques might be more suitable for different types of data and tasks, therefore, it is important to experiment with different techniques and see which ones work best for your specific case. Also, to evaluate the performance of the model with and without feature selection to make sure it is improving the performance and not degrading it.

The feature selection used fused vector V^k which includes a redundant feature for selecting the most discriminant features [73]. The proposed methodology utilized the concept of entropy-controlled mechanism [76]

, to analyze the uncertainty of data. Let F $V^K = (x1, t1),...,(xk, tk),...,(xN, tN)$ is a set of training data containing

N labels, where FV is a v-dimensional feature vector and $_j\underline{T}_1 = t_j^N$ are the class labels with tj belongs to the

number of classes. This feature space has φ used to measure the probability $\varphi(X)=1$, and the entropy calculated as:

$$E(X) = -\sum_{j=1}^{N}(x_j)\log\varphi(x_j) \qquad (1)$$

where φ(xj) is a probability for a particular feature xi. The basic functionality of the entropy mechanism to identify the unique/discriminant features. By using this mechanism, we assigned ranks to the feature V^E, having (R< N). All those random features are excluded from the resultant vector to select the most robust set of vectors. This rank based selection performs down sampling of the feature vector by keeping original information as it is and send it to the second stage of dimensional reduction.

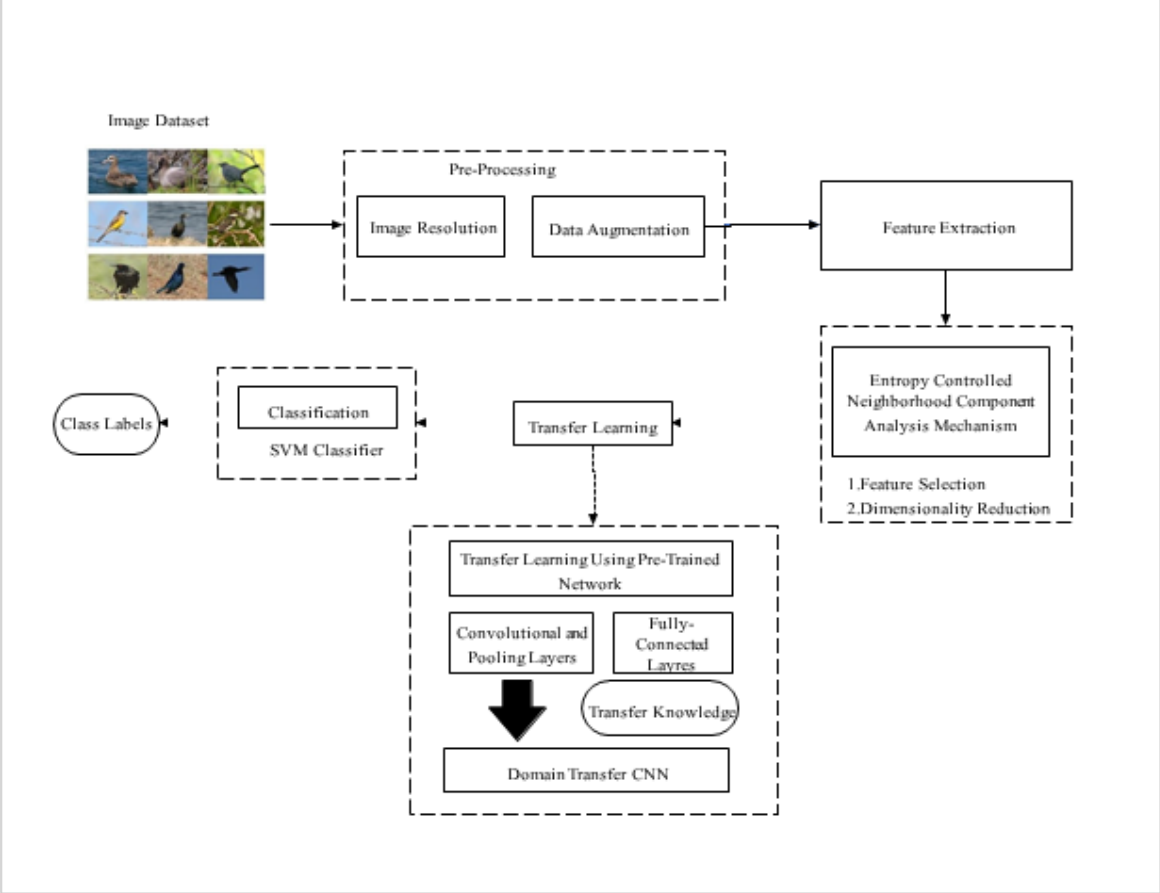


Figure 3: The proposed Entropy-Controlled Mechanism for Fine-Grained Image Classification

2 DIMENSIONAL REDUCTION

Dimensional reduction is a technique used to reduce the number of features in a dataset while preserving as much information as possible. The goal of dimensional reduction is to simplify the data while maintaining its important characteristics, which can improve the performance of machine learning models and make them more interpretable. There are several common dimensional reduction techniques that can be used:

5) Principal Component Analysis (PCA) is a linear technique that finds the most important directions (principal components) in the data and projects the data onto a lower-dimensional space along these directions.

6) Linear Discriminant Analysis (LDA) is a linear technique that finds the directions that maximize the separation between different classes in the data and projects the data onto a lower-dimensional space alongthese directions.

7) t-Distributed Stochastic Neighbor Embedding (t-SNE) is a non-linear technique that maps the data to a lower-dimensional space while preserving the local neighborhood structure of the data.

8) Autoencoder is a neural network-based technique that learns to compress the data into a lower-dimensional space and then reconstruct the original data.

9) UMAP is a non-linear technique that reduces the dimensionality of the data while preserving the global structure of the data, it's useful in visualizing high-dimensional data.

Different types of dimensional reduction techniques might be more suitable for different types of data and tasks,therefore, it's important to experiment with different techniques and see which ones work best for your specific case. Also, after reducing the dimensionality, it's important to make sure that the reduced dataset can still be used to train a good model and that the reduction doesn't lose too much information from the original data.

Dimensional reduction used to reduce the number of final vector and retain it in lower dimension. The Entropy controlled mechanism used for dimensional reduction by using Neighborhood Component Analysis (NCA) [75]. This function optimized the accuracy of a training data by using NCA technique. Therefore, the feature selectionand dimensional reduction helps in increasing classification accuracy.

The main objective is to learn a projection Q that maximized the accuracy of the data by defining a differential cost function based on neighbours in the transform space. Stating that data x_j select the neighbouring sample x_k as a reference probability, P_{jk}.

$$P_{jk} = \begin{cases} \frac{\gamma(-D(x_j, x_k))}{\sum_{j/=k} \gamma(-D(x_j, x_k))} & \text{if } i \neq k \\ 0, \quad otherwise \end{cases} \qquad (2)$$

Algorithm 1 Proposed Algorithm

Input: Set training data
Parameters: F V → feature vectors
F V^E → Entropy-Controlled Features
Q → Projection Matrix
D → Distance Matrix
P_{jk} → Mutual Probability
F V_{NCA} → Dimensional Reduce Features Vector
N → Total number of samples
Extract: Feature Vectors F V, compute F V , where k belongs to the number of classes
for i ={1 to L} do
Compute Entropy E(X) using Eq. (1)
Identify high ranked feature vectors to construct a new subspace FV^E
for i ={1,...., R} do
Initialize projection matrix Q = 0 = [1] ^{mx1}
Compute distance matrix D(x_j, x_k) using Eq. (??)
Compute assorted and labeling probabilities P^{jk}, P^{k} using Eq. (2)
for k ={ 1,...., d} do
Compute the cost function using eq & update projection matrix Q using conjugate gradient optimizer
$p = p + 1$
Repeat step 10 & 11 k times until convergence to find Q
Select subspace features F V_{NCA} based on projection matrix Q
Train:Selected Classifier SVM and Output: Class Labels

$$P_j = \sum_{k \in C_j} w_{jk} P_{jk} \quad (3)$$

The criteria used to search correct labels under one main category by using the following equation:

$$E(Q) = \sum_j \sum_i w_j P_{ij} = \sum_j P_j \quad (4)$$

Feature reduction approach used to avoid the over-fitting problem by introducing the regularization factor.

$h > 0$ in the cost function and few overcome via cross-validation [77], given as:

$$E(Q) = \sum_j \sum_i w_j P_{ij} - h \sum_{k=1}^{d} q_k^2 \quad (5)$$

D. TRANSFER LEARNING

Transfer learning concept is used in deep learning to obtain desirably results. It shows how to train a network on selected fine-grained dataset. Image resolution plays an important role in detection and classification accuracy. By using this method a pre-trained network used on a selected fine-grained dataset for classification. The image resolution used for this purpose is 331x331 to train a network for achieving tremendous performance in this specific domain. Traditional machine learning algorithms consider the same feature space for both training and testing dataset [78]. Due to this reason the concept of transfer learning was introduced which produces best results with limited or small amount of dataset. Few sub-categories in the fine-grained dataset have more number images, which causes imbalance. First, we must propose an effective way to address

these issues. In many trains with a massive number of training data through which the discriminative feature representation learned by network and in the second step fine-tuning the network on dataset contain balance and many images with minimum learning rate.

E. CLASSIFICATION

Image classification is the process of identifying and classifying objects, scenes, and other visual elements in an image. This is typically done using machine learning techniques, where a model is trained on a dataset of labeled images, and then used to classify new images. The model can be trained using various algorithms such as CNN (Convolutional Neural Network) and ResNet. The output of image classification is a label or set of labels that describe the content of the image.

Convolutional Neural Network used for detection and classification purposes [74]. In CNN's there is a feed forward and backward loop, in which the model can learn in a better way. The architecture of the Convolutional Neural Network is shown in the below Figure. 3, which includes convolutional layer, pooling layer and finally a fully connected layer for classification purpose. By increasing the number of layers, the number of parameters also goes increasing exponentially. To decrease the number of parameters used by the model for training purposes the concept of pooling layer was introduced. To reduce the number of parameters in all regions, pooling operation is performed followed by the convolutional layer. In this way the information is passed on in a feed forward manner from one layer to another.

IV. EXPERIMENTAL SETUP

This methodology was evaluated on the selected fine-grained dataset CUB-200-2011 [26] which is publicly available. Experimental setup with results on the selected dataset presented in Sec. IV.

DATA PREPARATION AND EXPERIMENTAL SETUP

A. DATASET

There is a wide variety of fine-grained datasets for the purpose of detection and classification. Examples of popular fine-grained datasets include Caltech-101,caltech256 [79, 80] and ImageNet [57]. The selected CUB-200-2011 [26] fine-grained dataset most used for birds' classification purpose. By using this selected dataset, it will help researchers in classification.

1) CUB-200-2011

In this research, simulations were done on the publicly available dataset. Caltech-UCSD Birds (CUB-2002011) is a crucial dataset annotated with 200 bird's categories, and the dataset contains a total of 11,788 images in which 5,994 used for training and 5,794 for testing purpose, as shown in detail in Table. 2.The dataset collected to facilitate the study of sub-classes which is not possible on any other dataset. Table.5 summarizes the results.

Table 2: Details of Fine-grained dataset

Dataset	CUB-200-2011
# Classes	200
Total Images	11,788
Training Images	5,994
Testing Images	5,794

B. NETWORK ARCHITECTURE

We used NASNet Large network architecture in our proposed method.

1) NASNet Large

NasNet is a convolutional neural network (CNN) architecture for image classification tasks. It was introduced in the paper [60] by Google Brain team in 2017. The key feature of NasNet is its use of neural architecture search (NAS) to automatically search for the best CNN architecture for a given task. This contrasts with traditional CNNs, which are designed by hand. NasNet uses a combination of convolutional and recurrent neural networks to learn the architecture, and it is trained on a large dataset of images. NASNet Large is a specific variant of the NASNet Large architecture designed for large-scale image classification tasks. It was trained on the ImageNet dataset [11] and can achieve state-of-the-art accuracy on this benchmark. For bird classification using the CUB-200-2011 dataset, NASNet Large could be fine-tuned on this dataset. Fine-tuning is the process of training a pre-trained model on a new dataset, where the model's parameters are initialized with the pre-trained weights, and then the model is trained on the new dataset. This process allows the model to learn the specific features of the new dataset, while still utilizing the knowledge learned from the original dataset. It's worth noting that NASNet Large is a very large model and fine-tuning will require a lot of computational power and a large amount of data to get good results.

To achieve state-of-the-art accuracy on the ImageNet dataset [11], which is a large dataset of images used for image classification tasks. The authors of the paper [60] reported that NasNet-Large achieved a top-1 accuracy of 82.7% and a top-5 accuracy of 96.2% on the ImageNet validation set. In comparison to other models at the time, NasNet-Large outperformed the previous state-of-the-art model, Inception-v3 [69], by 0.7% in top-1 accuracy and 0.2% in top-5 accuracy. It's also worth noting that NasNet-Large is a very large model with 88.9 million parameters, which is significantly larger than Inception-v3 [69] (23.8 million parameters). It is important to note that the accuracy of a model on a specific task depends on the quality and quantity of data, the fine- tuning technique, and the specific application. So, the accuracy of NasNet-Large on other datasets or tasks might be different. As mentioned earlier, the concept of transfer learning is used for the purpose of classification. The network NASNet Large pre-trained on ImageNet dataset [11] which contain almost 1k categories. The network can be able to learn more features easily by using an image resolution of 331x331 by default. The challenge of the selected model is the non-linearity introduced by the convolution filters and the carefull selection of the connection between the layers to gain impressive performance. NASNet architecture consists of many normal and reduction cells. After knowing about the architecture flow, a few parameters must be analyzed to build a network for the specific task.

1) The total number of cells in the network repeats the N.

2) And the total number of filters used in the network.

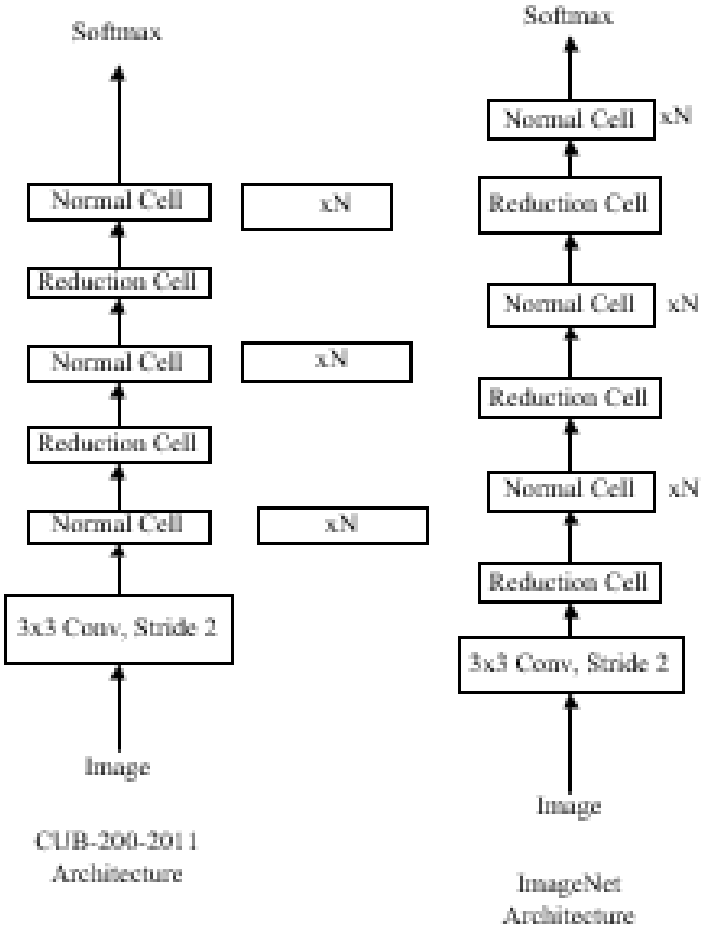


Figure 4: The CUB-200-2011 Architecture

C. IMPLEMENTATION DETAILS

While training a model, by using a high-resolution input image 331x331 and a Scheduled Dropout Path the newer version of the Dropout [60] which is an efficient regularization parameter. By applying these two modifications the performance also increases. The initial learning rate is $1e^{-5}$, while the weight decay and momentumare 0.5 and 0.9, respectively. The batch size of 64 for 300 epochs is enough for convergence and after 50 iterations it reached stable condition. Experiments were implemented by open-source TensorFlow [81] and Keras to train and test the model on NVIDIA GeForce GTX 1080 Ti GPU. The accuracy parameter used to evaluate the fine-grained image classification performance.

1) Equipment

The TensorFlow and Keras as a framework used in our research, OpenCV as a development environment and Python as a language used for training and testing the pre-trained model on a GPU. The other parameters are shown below in Table. 3.

Table 3: System Parameters used for the implementation of the network.

Name	Parameters
Memory (RAM)	32GB
CPU-Processor	Intel Core i7-8730 CPU
	@3.43GHz 3.413GHz

Graphics	NVIDIA GeForce GTX 1080 Ti
Operating system	Windows-10 64 bits
CUDA Toolkit	11.0
Development environment	Tensorflow & Keras
Image Processing Library	OpenCv
Language	Python

D. TRANSFER LEARNING BASED FINE-GRAINED IMAGE CLASSIFICATION

Transfer learning is a technique that allows a pre-trained model, such as NasNet-Large, to be fine-tuned on a new dataset for a specific task. This process involves using the pre-trained model's parameters as a starting point, and then training the model on the new dataset. To fine-tune NasNet-Large for image classification on a new dataset, the following steps can be taken:

(i) Initialize the model's parameters with the pre-trained weights.

(ii) Replace the last fully connected layer of the model with a new layer that corresponds to the number of classes in the new dataset.

(iii) Freeze the rest of the layers (except the last one) during the fine-tuning process, so that the model's pre-trained features are not modified.

(iv) Train the model on the new dataset using a suitable optimizer and loss function.

(v) Fine-tune the model by adjusting the learning rate and the number of training epochs, until the desiredlevel of accuracy is achieved.

Transfer learning is a powerful technique and can be used to train a model on a small dataset. However, it's important to have a large and diverse dataset for fine-tuning to achieve good accuracy. Also, it's important to monitor the performance of the model during the fine-tuning process and adjust as necessary. To evaluate the proposed methodology, we must perform experiments on the selected CUB-200-2011 [26] dataset.

To obtain impressive performance, used the concept of transfer learning on 331x331 resolution images with a minimum learning rate of $10e^{-5}$ in our experiments.

Table 4: Transfer learning results on CUB-200-2011 by using NasNet network 331x331 pre-trained on ImageNet.

	Dataset	**Feature Vector**	**Overall Accuracy % Proposed (ECNCA)**
ImageNet	CUB-200-2011	FV0-FV1	83.7
		FV0-FV2	82.6
		FV1-FV2	80.2
		FV0-FV1-FV2	92.6

ImageNet achieved different transfer learning performance on the target dataset as shown below in Table. 4. The selected data set also achieved the best performance on the ImageNet pre-trained network, as shown below in Table.5. In the previous knowledge work in this domain, there are few options to achieve better accuracyon fine-grained datasets by using the deepest model based on standard datasets [41, 82] with

augmentation techniques [53]. An approach to use a pre-trained network on a selected dataset to increase its performance by transfer learning method. We performed simulation on the publicly available dataset, as shown in Table. 2.

The methodology evaluated using three simulation steps—the classification results obtained from selected layers of the pre-trained model. The second simulation step used the NCA technique in combination with the

proposed feature reduction approach. While in the third step, tested the proposed technique using state-of-the-artSVM classifier.

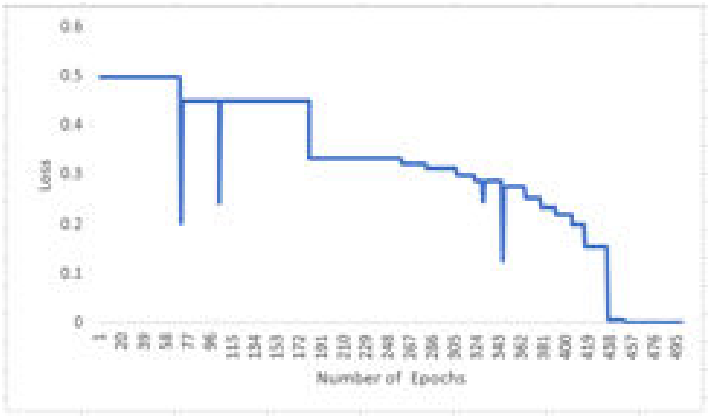


Figure 5: Loss Graph

1) Evaluation of proposed technique

Features extracted from the previous layers and then concatenated by using the feature selection and reduction steps. The maximum reduction percentage achieved is 92.6% on the CUB-200-2011 dataset. We created four combination feature vectors from the dataset. The regularization parameter, such as λ (kernel width), selected to be0:0039 following a trial-and-error approach. By using entropy-controlled feature fusion approach, on the chosen dataset, gives maximum accuracy of 92.6% using SVM classifier on feature vectors FV0-FV1-FV2.

E. COMPARISON WITH STATE-OF-THE-ART METHODS

Results show that the best classification results on a given dataset are achieved using the proposed methodology. The maximum classification accuracy achieved on CUB-200-2011 is 90.4% by using Stacked LSTM [83], and by using this proposed methodology the accuracy is 92.6%. Comparison with existing techniques given in Table.5 on selected dataset.

Table 5: Comparison with the existing techniques on selected dataset.

Method	CUB-200-2011
VGG -19 [66]	77.8%
ResNet -101 [67]	83.5%
Inception -V3 [69]	82.7%
Billinear -CNN [41]	84.1%
ST- CNN [59]	84.1%
PDFR [21]	84.5%
RA -CNN [42]	85.4%
MA -CNN [84]	86.5%

GP -256 [85]	85.8%
MAMC [86]	86.5%
PC [87]	86.9%
DFL-CNN [88]	87.4%
NTS -Net [89]	87.5%
MPN -COV [90]	88.7%
Inception -ResNet -v2-SE [38]	89.3%
WS-DAN [91]	89.4%
baseline + Stacked LSTM + Multi-Loss [83]	90.4%
Our Proposed	92.6%

F. VISUALIZATION OF AUGMENTED DATA

In Figure.4, CUB-200-2011 dataset augmented by using randomly different augmentation techniques. Intuitively, the data augmentation introduced background images in the dataset as well as it also helps in increasing the number of images for training purposes. In data augmentation different techniques like cropping, scaling, rotation, contrast enhancement, flipping and brightness applied. Also, validate these augmentation techniques on the validation dataset.

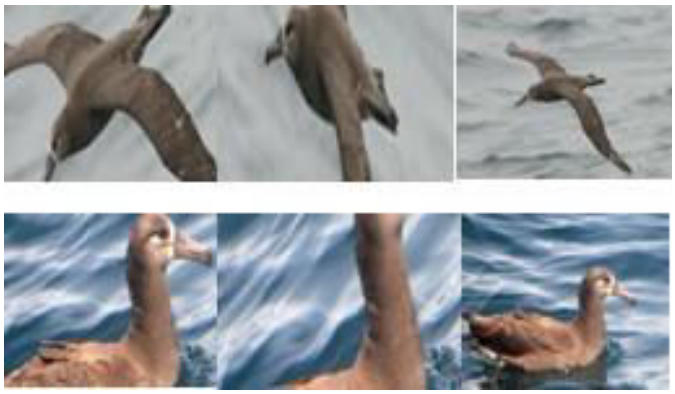


Figure 6: Visualization of augmented images of random data augmentation.

V. CONCLUSION

In this work, evaluated the model how it learned from training data by using the concept of transfer learning for fine-grained image classification task. Further proposed an Entropy-controlled mechanism which uses a hierarchical framework used for discriminant feature selection and dimensional reduction. By exploiting extracted information from pre-trained model by using the concept of transfer learning, which significantly contributes to the enhancement of overall accuracy. Also, by utilizing less amount of the total features, it helps in removing redundancy and minimizing computational time. After doing all these experiments, results concluded as; a) the fused extracted feature from a pre-trained model improves accuracy b) Feature selection and dimensional reduction also helps in improving classification results as well as in reducing computational time. As future work, we can apply this technique to more complex datasets for a comprehensive comparison study of other significant factors.

ACKNOWLEDGMENT

Research work supported by the Computer Engineering Department of the University of Engineering & Technology Taxila.

REFERENCES

[1] J. Redmon and A. Angelova, 2015 IEEE International Conference on Robotics and Automation (ICRA), 2015,pp. 1316–1322.

[2] D. Michie, D. J. Spiegelhalter, C. Taylor et al., Neural and Statistical Classification, 1994, **13**, 1–298.

[3] H. Lee, R. Grosse, R. Ranganath and A. Y. Ng, Proceedings of the 26th annual international conference onmachine learning, 2009, pp. 609–616.

[4] Y. LeCun, Y. Bengio and G. Hinton, nature, 2015, **521**, 436–444.

[5] C. Zhao, C. Liu and Z. Lai, Neurocomputing, 2011, **74**, 2929–2940.

[6] R. Nosaka, Y. Ohkawa and K. Fukui, Pacific-Rim Symposium on Image and Video Technology, 2011, pp.82–91.

[7] P. E. Rybski, D. Huber, D. D. Morris and R. Hoffman, 2010 IEEE Intelligent vehicles symposium, 2010, pp.921–928.

[8] P. C. Ng and S. Henikoff, Nucleic acids research, 2003, **31**, 3812–3814.

[9] H. Chang, J. Han, C. Zhong, A. M. Snijders and J.-H. Mao, IEEE transactions on pattern analysis and machineintelligence, 2017, **40**, 1182–1194.

[10] A. G. Howard, arXiv preprint arXiv:1312.5402, 2013.

[11] A. Krizhevsky, I. Sutskever and G. E. Hinton, Advances in neural information processing systems, 2012, pp.1097–1105.

[12] A. Sharif Razavian, H. Azizpour, J. Sullivan and S. Carlsson, Proceedings of the IEEE conference on computervision and pattern recognition workshops, 2014, pp. 806–813.

[13] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng and T. Darrell, International conference onmachine learning, 2014, pp. 647–655.

[14] N. Zhang, J. Donahue, R. Girshick and T. Darrell, European conference on computer vision, 2014, pp. 834–849.

[15] Z. Ge, C. McCool, C. Sanderson and P. Corke, 2015 IEEE International Conference on Image Processing(ICIP), 2015, pp. 4112–4116.

[16] K. He, X. Zhang, S. Ren and J. Sun, Proceedings of the IEEE international conference on computer vision,2015, pp. 1026–1034.

[17] S. Cai, W. Zuo and L. Zhang, Proceedings of the IEEE International Conference on Computer Vision, 2017,pp. 511–520.

[18] S. Kong and C. Fowlkes, Proceedings of the IEEE conference on computer vision and pattern recognition,2017, pp. 365–374.

[19] X. He and Y. Peng, arXiv preprint arXiv:1709.00340, 2017.

[20] M. Lam, B. Mahasseni and S. Todorovic, Proceedings of the IEEE Conference on Computer Vision andPattern Recognition, 2017, pp. 2520–2529.

[21] X. Zhang, H. Xiong, W. Zhou, W. Lin and Q. Tian, IEEE Transactions on Multimedia, 2017, **19**, 2736–2750.

[22] B. Hariharan, P. Arbeláez, R. Girshick and J. Malik, European Conference on Computer Vision, 2014, pp.297–312.

[23] R. Girshick, J. Donahue, T. Darrell and J. Malik, Proceedings of the IEEE conference on computer vision andpattern recognition, 2014, pp. 580–587.

[24] J. Yosinski, J. Clune, Y. Bengio and H. Lipson, Advances in neural information processing systems, 2014, pp.3320–3328.

[25] T. Berg and P. Belhumeur, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,2013, pp. 955–962.

[26] C. Wah, S. Branson, P. Welinder, P. Perona and S. Belongie, 2011.

[27] A. Khosla, N. Jayadevaprakash, B. Yao and F.-F. Li, Proc. CVPR Workshop on Fine-Grained VisualCategorization (FGVC), 2011.

[28] L. Yang, P. Luo, C. Change Loy and X. Tang, Proceedings of the IEEE Conference on Computer Vision andPattern Recognition, 2015, pp. 3973–3981.

[29] Y. Chai, V. Lempitsky and A. Zisserman, Proceedings of the IEEE International Conference on ComputerVision, 2013, pp. 321–328.

[30] R. Farrell, O. Oza, N. Zhang, V. I. Morariu, T. Darrell and L. S. Davis, 2011 International Conference onComputer Vision, 2011, pp. 161–168.

[31] E. Gavves, B. Fernando, C. G. Snoek, A. W. Smeulders and T. Tuytelaars, Proceedings of the IEEE internationalconference on computer vision, 2013, pp. 1713–1720.

[32] R. H. Zottesso, Y. M. Costa, D. Bertolini and L. E. Oliveira, Ecological Informatics, 2018, **48**, 187–197.

[33] S. Branson, P. Perona and S. Belongie, 2011 International Conference on Computer Vision, 2011, pp. 1832–1839.

[34] J. Liu, A. Kanazawa, D. Jacobs and P. Belhumeur, European conference on computer vision, 2012, pp. 172–185.

[35] N. Zhang, R. Farrell, F. Iandola and T. Darrell, Proceedings of the IEEE International Conference on ComputerVision, 2013, pp. 729–736.

[36] S. Branson, G. Van Horn, S. Belongie and P. Perona, arXiv preprint arXiv:1406.2952, 2014.

[37] Z. Ge, C. McCool, C. Sanderson and P. Corke, Proceedings of the IEEE Conference on Computer Vision andPattern Recognition Workshops, 2015, pp. 46–52.

[38] Y. Cui, Y. Song, C. Sun, A. Howard and S. Belongie, Proceedings of the IEEE conference on computer visionand pattern recognition, 2018, pp. 4109–4118.

[39] N. Ammour, H. Alhichri, Y. Bazi, B. Benjdira, N. Alajlan and M. Zuair, Remote Sensing, 2017, **9**, 312.

[40] S.-J. Hong, Y. Han, S.-Y. Kim, A.-Y. Lee and G. Kim, Sensors, 2019, **19**, 1651.

[41] T.-Y. Lin, A. RoyChowdhury and S. Maji, Proceedings of the IEEE international conference on computervision, 2015, pp. 1449–1457.

[42] J. Fu, H. Zheng and T. Mei, Proceedings of the IEEE conference on computer vision and pattern recognition,2017, pp. 4438–4446.

[43] T. Xiao, Y. Xu, K. Yang, J. Zhang, Y. Peng and Z. Zhang, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 842–850.

[44] F. Schroff, D. Kalenichenko and J. Philbin, Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 815–823.

[45] Y. Cui, F. Zhou, Y. Lin and S. Belongie, Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 1153–1162.

[46] N. Zhang, E. Shelhamer, Y. Gao and T. Darrell, arXiv preprint arXiv:1511.07063, 2015.

[47] T. Gebru, J. Hoffman and L. Fei-Fei, Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1349–1358.

[48] A. Vedaldi, S. Mahendran, S. Tsogkas, S. Maji, R. Girshick, J. Kannala, E. Rahtu, I. Kokkinos, M. B. Blaschko,

D. Weiss et al., Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 3622–3629.

[49] S. Branson, C. Wah, F. Schroff, B. Babenko, P. Welinder, P. Perona and S. Belongie, European Conference on Computer Vision, 2010, pp. 438–451.

[50] J. Deng, J. Krause, M. Stark and L. Fei-Fei, IEEE transactions on pattern analysis and machine intelligence, 2015, **38**, 666–676.

[51] S. Reed, Z. Akata, H. Lee and B. Schiele, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 49–58.

[52] X. He and Y. Peng, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5994–6002.

[53] J. Krause, B. Sapp, A. Howard, H. Zhou, A. Toshev, T. Duerig, J. Philbin and L. Fei-Fei, European Conference on Computer Vision, 2016, pp. 301–320.

[54] Z. Xu, S. Huang, Y. Zhang and D. Tao, IEEE transactions on pattern analysis and machine intelligence, 2016, **40**, 1100–1113.

[55] X. Zhu, D. Anguelov and D. Ramanan, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 915–922.

[56] G. Van Horn and P. Perona, arXiv preprint arXiv:1709.01450, 2017, 7132–7141.

[57] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, 2009 IEEE conference on computer vision and pattern recognition, 2009, pp. 248–255.

[58] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár and C. L. Zitnick, European conference on computer vision, 2014, pp. 740–755.

[59] M. Jaderberg, K. Simonyan, A. Zisserman et al., Advances in neural information processing systems, 2015, pp. 2017–2025.

[60] B. Zoph, V. Vasudevan, J. Shlens and Q. V. Le, Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 8697–8710.

[61] A. E. Öztürk and E. Erçelebi, Applied Sciences, 2021, **11**, 3863.

[62] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva and A. Torralba, IEEE transactions on pattern analysis and machine intelligence, 2017, **40**, 1452–1464.

[63] C. Sun, A. Shrivastava, S. Singh and A. Gupta, Proceedings of the IEEE international conference on computer vision, 2017, pp. 843–852.

[64] M. Huh, P. Agrawal and A. A. Efros, arXiv preprint arXiv:1608.08614, 2016.

[65] H. Azizpour, A. S. Razavian, J. Sullivan, A. Maki and S. Carlsson, IEEE transactions on pattern analysis and machine intelligence, 2015, **38**, 1790–1802.

[66] K. Simonyan and A. Zisserman, arXiv preprint arXiv:1409.1556, 2014.

[67] K. He, X. Zhang, S. Ren and J. Sun, Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[68] C. Szegedy, S. Ioffe, V. Vanhoucke and A. A. Alemi, Thirty-First AAAI Conference on Artificial Intelligence, 2017.

[69] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2818–2826.

[70] K. He, X. Zhang, S. Ren and J. Sun, European conference on computer vision, 2016, pp. 630–645.

[71] S. Xie, R. Girshick, P. Dollár, Z. Tu and K. He, Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1492–1500.

[72] J. Hu, L. Shen and G. Sun, Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 7132–7141.

[73] T. Akram, B. Laurent, S. R. Naqvi, M. M. Alex, N. Muhammad et al., Information Sciences, 2018, **467**, 199–218.

[74] D. Xie, L. Zhang and L. Bai, Applied Computational Intelligence and Soft Computing, 2017, **2017**, 343–359.

[75] J. Goldberger, G. E. Hinton, S. T. Roweis and R. R. Salakhutdinov, Advances in neural information processing systems, 2005, pp. 513–520.

[76] A. S. Sankar, S. S. Nair, V. S. Dharan and P. Sankaran, Procedia Computer Science, 2015, **46**, 1476–1482.

[77] W. Yang, K. Wang and W. Zuo, JCP, 2012, **7**, 161–168.

[78] K. Weiss, T. M. Khoshgoftaar and D. Wang, Journal of Big data, 2016, **3**, 9.

[79] L. Fei-Fei, R. Fergus and P. Perona, IEEE transactions on pattern analysis and machine intelligence, 2006, **28**, 594–611.

[80] G. Griffin, A. Holub and P. Perona, 2007.

[81] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard et al., 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), 2016, pp. 265–283.

[82] Y. Cui, F. Zhou, J. Wang, X. Liu, Y. Lin and S. Belongie, Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2921–2930.

[83] W. Ge, X. Lin and Y. Yu, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 3034–3043.

[84] H. Zheng, J. Fu, T. Mei and J. Luo, Proceedings of the IEEE international conference on computer vision,2017, pp. 5209–5217.

[85] X. Wei, Y. Zhang, Y. Gong, J. Zhang and N. Zheng, Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 355–370.

[86] M. Sun, Y. Yuan, F. Zhou and E. Ding, Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 805–821.

[87] A. Dubey, O. Gupta, P. Guo, R. Raskar, R. Farrell and N. Naik, Proceedings of the European Conference onComputer Vision (ECCV), 2018, pp. 70–86.

[88] Y. Wang, V. I. Morariu and L. S. Davis, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4148–4157.

[89] Z. Yang, T. Luo, D. Wang, Z. Hu, J. Gao and L. Wang, Proceedings of the European Conference on ComputerVision (ECCV), 2018, pp. 420–435.

[90] P. Li, J. Xie, Q. Wang and Z. Gao, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 947–955.

[91] T. Hu and H. Qi, arXiv preprint arXiv:1901.09891, 2019.