2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD) | 978-1-6654-1460-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICICT4SD50815.2021.9396917

High Performance Classification of Caltech-101 with a Transfer Learned Deep Convolutional Neural Network

Md. Mehedi Hasan*, Azmain Yakin Srizon†, Abu Sayeed‡ and Md. Al Mehedi Hasan§
Department of Computer Science & Engineering
Rajshahi University of Engineering & Technology, Rajshahi, Bangladesh
Email: *mmehedihasann@gmail.com, †azmainsrizon@gmail.com, ‡abusayeed.cse@gmail.com, §mehedi_ru@yahoo.com

***Abstract*—Numerous models and working schemes have been proposed through decades for the successful recognition of the objects. Significant contributions are notable in the field of object recognition. However, near accurate recognition is still a challenge in this domain. In this research, we considered the Caltech-101 dataset having 102 diverse and imbalanced classes i.e., people, animals, landscapes, structures, furniture, etc. which made the recognition more complicated. We proposed and utilized modified InceptionV3 and modified EfficientNetB6 architectures for the recognition of objects which obtained 99.65% and 99.72% overall accuracy respectively. We further showed via experimental analysis that the softmax-averaging technique can further boost the accuracy to 99.85% and all three proposed procedures suppressed the previous studies by a notable boundary as well.**

***Index Terms*—Object Recognition, Caltech-101, Deep Convolutional Neural Network, InceptionV3, EfficientNetB6, Softmax-averaging, Augmentation**

I. Introduction

Object recognition has always been an essential and integrated part of computer vision and digital image processing that deals with identifying various groups, for example, faces, structures, animals, and many other natural and human-made objects in digital images or videos [1]. In a fast-paced world, object recognition can be utilized for numerous aspects. Movement perception via video surveillance is one of the most common tasks of object recognition where multiple people or objects can be tracked or traced at once in real-time [2]. Another rising industry for object recognition in the computer vision domain is pictorial explanation where information can be retrieved or comments can be generated based on the objects detected in the scene [3]. Object recognition is also widely utilized in the sports industry. Almost all popular sports i.e., cricket, football, baseball, basketball, etc. utilize object recognition for tracking players, sports equipment, and other essential elements for providing a smooth and satisfying experience to the spectators. Moreover, face recognition is one of the basic examples of object recognition where faces can be recognized by utilizing shape-based features i.e., positions of eyes, nose, and mouth. Some other important areas related to object recognition are crowd counting, anomaly detection, self-driving cars, materials detection, robotics, automated CCTV, etc. Despite having numerous applications of object recognition, detecting all these different types of objects by a single model is a challenging task. Face, animals, structures, landscapes, furniture, etc. are completely different in shape and nature than one another which makes the generalization task of detecting all sorts of objects more complicated.

In this research, the focus has been provided on the challenge of detecting multiple types of objects. The Caltech-101 dataset has been considered in this work that consists of 102 different classes. These classes involve people, animals, landscapes, structures, furniture, natural objects, human-made objects, and so on. The objective of the research was to design a deep convolutional neural network model that is capable of near-accurate recognition. Previously, many kinds of research have been conducted to find a suitable model but the performance of the classifiers was not satisfactory. In this work, modified inceptionV3 and modified efficientNetB6 architectures have been utilized for the recognition of 102 groups. The experimental analysis showed that both of the architectures are capable of producing a high classification accuracy of 99.65% and 99.72% respectively. Also, it was revealed that softmax-averaging of two classifiers boosted the accuracy even more to 99.85% which outperformed the previous best accuracy of 94.38% achieved by VGG-16.

II. Literature Review

Object recognition problem has been revisited by the researchers several times through decades because of the development of newly formed datasets, and discovery of machine learning [4], [5] and deep learning algorithms [6], [7]. The Caltech-101 dataset has been also revisited multiple times throughout the last decade because of its complexity. One of the very first attempts of Caltech-101 object recognition was done by Lee et al. in 2009 when the authors proposed a convolutional neural network-based technique to identify 102 objects and obtained 65.4% overall accuracy [8]. However, not many contributions have been done since then till 2018. But after the machine-boost era and the update of the dataset, the dataset was revisited by researchers. One of the impactful work was proposed by Song et al. who achieved 83.9% overall accuracy by utilizing principal component analysis on SIFT characteristics in 2018 [9]. Li et al. proposed the EL+YcbCr technique in 2018 as well but obtained 78% overall accuracy

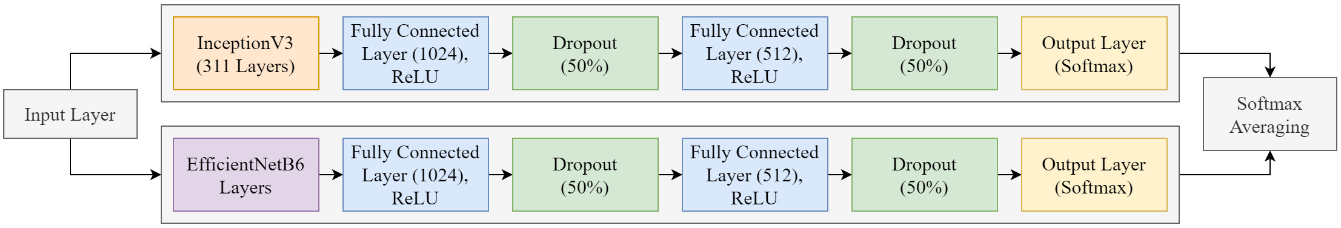

Fig. 1: Modified InceptionV3 Architecture (Upper Part), Modified EfficientNetB6 Architecture (Lower Part) and Softmax-averaging Technique (Overall Picture).

[10]. In the meantime, Pan et al. introduced a k-mean reduction mechanism and utilized a convolutional neural network which boosted the overall accuracy to 85.78% [11]. However, in 2019, Cubuk et al. introduced augmentation and proved that it can further boost the overall accuracy by obtaining 86.9% accuracy [12]. Also, in 2019, Rashid et al. reinvestigated the SIFT based features but this time by utilizing a convolutional neural network which produced 89.7% accuracy [13]. Later that year, another study reported 91.8% overall accuracy as well [14]. In 2020, a study of different models such as VGG-16, ResNet-50, MobileNet, DenseNet-121 and NASNetMobile was published where the authors reported 94.38%, 91.13%, 92.07%, 89.5% and 87.77% overall accuracy respectively [15]. Furthermore, during the same year, Hussain et al. suggested a deep neural network and classical features based scheme for object recognition achieving 90.1% overall accuracy [16].

III. Materials and Methods

A. Dataset Description

The Caltech-101 dataset was taken under consideration while conducting our research work which had 102 distinct pictorial groups or classes [17]. The total number of samples was 9,145 and the dataset was imbalanced. The diversity of the dataset was up to the mark because of the variation in image sizes and types i.e., grayscale and RGB.

B. Convolutional Neural Network

The convolutional neural networks, typically signified by CNN, is one of the most common aspects of deep learning modules involving convolution, pooling, and fully connected layers [18]. Throughout decades, the convolutional neural network has been utilized for medical image recognition, natural language processing, IoT, self-driving cars and financial data analysis. Convolution layers typically extract valuable features from the input data by finding out the best filters. Pooling layers are utilized for reshaping after applying convolutions and the fully connected layers serve the service of multilayer perceptrons. Since CNN doesn't need additional feature engineering it is considered one of the powerful tools for image classification.

C. Transfer Learning

Transfer learning refers to the phenomenon of utilizing previously trained weights to solve a completely new but somehow similar dilemma [19]. For example, the ImageNet dataset contains 1,000 different groups. Some of these groups are extremely similar while others are completely different from one another. That's why the ImageNet challenge weights will be a suitable start for any recognition which involves similar or different groups. For instance, the classifier that can classify different types of cars can also classify trucks with some modifications on the previous weights.

D. Modified InceptionV3 Architecture

InceptionV3 started its journey as a part of the Googlenet [20] for solving the ImageNet challenge. As the name suggests, it was the third version of Google's inception CNN architecture that consists of a total of 311 layers. In this research, we utilized all those 311 layers of the InceptionV3 architecture. However, after that, we added two fully connected layers of size 1024 and 512 followed by a dropout of 50% for each of the fully connected layers. Freezing the layers was ignored while implementing the InceptionV3 network.

E. Modified EfficientNetB6 Architecture

As model scaling does not modify layer operators in the baseline network, producing a better baseline network is also complex. To adequately illustrate the effectiveness of the scaling method, a new mobile-size baseline, called EfficientNet was constructed [21]. Inspired by [22], it was introduced by a baseline network by leveraging a multi-objective neural structure search that optimizes both accuracy and FLOPS. Therefore, the authors optimized FLOPS despite latency as they were not targeting any definite hardware design. Here, we utilized EfficientNetB6 architecture and added two additional fully connected layers of size 1024 and 512 followed by 50% dropout for each. Freezing the layers was ignored for EfficientNetB6 implementation as well.

F. Softmax-averaging Mechanism

While applying the output layers for both InceptionV3 and EfficientNetB6 architecture, the softmax activation function was utilized. The softmax activation function produces a probability of recognition for each of the classes. Softmax-averaging refers to the technique of averaging the softmax values produced by multiple classifiers and consider the average matrix as the final outcome. This technique can boost the overall performance if the classifiers are getting confused between two classes that are too close to call. For example, suppose one sample from class B produced softmax values

TABLE I: Class-wise accuracy, precision, recall, and f1-score (F1.) for softmax-averaging mechanism

Classes	Accuracy	Precision	Recall	F1.	Classes	Accuracy	Precision	Recall	F1.
Background	0.86	1.00	0.86	0.92	helicopter	1.00	1.00	1.00	1.00
Faces	1.00	1.00	1.00	1.00	ibis	1.00	1.00	1.00	1.00
Faces_easy	1.00	1.00	1.00	1.00	inline_skate	1.00	1.00	1.00	1.00
Leopards	1.00	1.00	1.00	1.00	joshua_tree	1.00	1.00	1.00	1.00
Motorbikes	1.00	1.00	1.00	1.00	kangaroo	1.00	1.00	1.00	1.00
accordion	1.00	0.99	1.00	1.00	ketch	1.00	1.00	1.00	1.00
airplanes	1.00	1.00	1.00	1.00	lamp	1.00	1.00	1.00	1.00
anchor	1.00	1.00	1.00	1.00	laptop	1.00	1.00	1.00	1.00
ant	1.00	1.00	1.00	1.00	llama	1.00	1.00	1.00	1.00
barrel	1.00	1.00	1.00	1.00	lobster	1.00	1.00	1.00	1.00
bass	1.00	1.00	1.00	1.00	lotus	1.00	1.00	1.00	1.00
beaver	1.00	1.00	1.00	1.00	mandolin	1.00	1.00	1.00	1.00
binocular	1.00	0.99	1.00	1.00	mayfly	1.00	1.00	1.00	1.00
bonsai	1.00	0.99	1.00	1.00	menorah	1.00	1.00	1.00	1.00
brain	1.00	1.00	1.00	1.00	metronome	1.00	1.00	1.00	1.00
brontosaurus	1.00	1.00	1.00	1.00	minaret	1.00	1.00	1.00	1.00
buddha	1.00	1.00	1.00	1.00	nautilus	1.00	1.00	1.00	1.00
butterfly	1.00	1.00	1.00	1.00	octopus	1.00	1.00	1.00	1.00
camera	1.00	1.00	1.00	1.00	okapi	1.00	1.00	1.00	1.00
cannon	1.00	1.00	1.00	1.00	pagoda	1.00	1.00	1.00	1.00
car_side	1.00	1.00	1.00	1.00	panda	1.00	0.99	1.00	1.00
ceiling_fan	1.00	1.00	1.00	1.00	pigeon	1.00	1.00	1.00	1.00
cellphone	1.00	0.99	1.00	1.00	pizza	1.00	0.98	1.00	0.99
chair	1.00	0.99	1.00	1.00	platypus	1.00	1.00	1.00	1.00
chandelier	1.00	1.00	1.00	1.00	pyramid	1.00	1.00	1.00	1.00
cougar_body	1.00	1.00	1.00	1.00	revolver	0.99	1.00	0.99	0.99
cougar_face	1.00	1.00	1.00	1.00	rhino	1.00	1.00	1.00	1.00
crab	1.00	0.97	1.00	0.99	rooster	1.00	1.00	1.00	1.00
crayfish	1.00	1.00	1.00	1.00	saxophone	1.00	1.00	1.00	1.00
crocodile	1.00	1.00	1.00	1.00	schooner	1.00	1.00	1.00	1.00
crocodile_head	1.00	1.00	1.00	1.00	scissors	1.00	1.00	1.00	1.00
cup	1.00	1.00	1.00	1.00	scorpion	1.00	0.99	1.00	1.00
dalmatian	1.00	1.00	1.00	1.00	sea_horse	1.00	1.00	1.00	1.00
dollar_bill	1.00	1.00	1.00	1.00	snoopy	1.00	1.00	1.00	1.00
dolphin	1.00	1.00	1.00	1.00	soccer_ball	1.00	1.00	1.00	1.00
dragonfly	1.00	1.00	1.00	1.00	stapler	1.00	1.00	1.00	1.00
electric_guitar	1.00	1.00	1.00	1.00	starfish	1.00	1.00	1.00	1.00
elephant	1.00	1.00	1.00	1.00	stegosaurus	1.00	1.00	1.00	1.00
emu	1.00	0.99	1.00	1.00	stop_sign	1.00	0.99	1.00	1.00
euphonium	1.00	1.00	1.00	1.00	strawberry	1.00	1.00	1.00	1.00
ewer	1.00	1.00	1.00	1.00	sunflower	1.00	1.00	1.00	1.00
ferry	1.00	1.00	1.00	1.00	tick	1.00	1.00	1.00	1.00
flamingo	1.00	1.00	1.00	1.00	trilobite	1.00	1.00	1.00	1.00
flamingo_head	1.00	1.00	1.00	1.00	umbrella	1.00	1.00	1.00	1.00
garfield	1.00	0.99	1.00	1.00	watch	1.00	1.00	1.00	1.00
gerenuk	1.00	1.00	1.00	1.00	water_lilly	1.00	1.00	1.00	1.00
gramophone	1.00	1.00	1.00	1.00	wheelchair	1.00	1.00	1.00	1.00
grand_piano	1.00	1.00	1.00	1.00	wild_cat	1.00	1.00	1.00	1.00
hawksbill	1.00	1.00	1.00	1.00	windsor_chair	1.00	1.00	1.00	1.00
headphone	1.00	1.00	1.00	1.00	wrench	1.00	1.00	1.00	1.00
hedgehog	1.00	1.00	1.00	1.00	yin_yang	1.00	1.00	1.00	1.00

of [0.6, 0.4] and [0.1, 0.9] for two classifiers when two classes, A and B, are under consideration. The first classifier predicted class A where the second classifier predicted class B. But by taking an average of two we obtain [0.35, 0.65] which indicates class B. Hence, when two classes are too close to call, softmax-averaging can boost the performance significantly even if only one of the classifiers produced the correct outcome. Fig. 1 shows the modified InceptionV3, modified EfficientNetB6 architectures and Softmax-averaging mechanism.

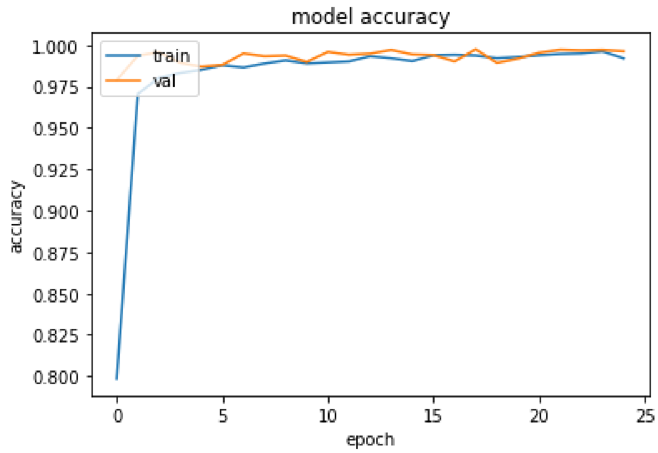

Fig. 2: Training accuracy and validation accuracy of proposed modified InceptionV3 architecture

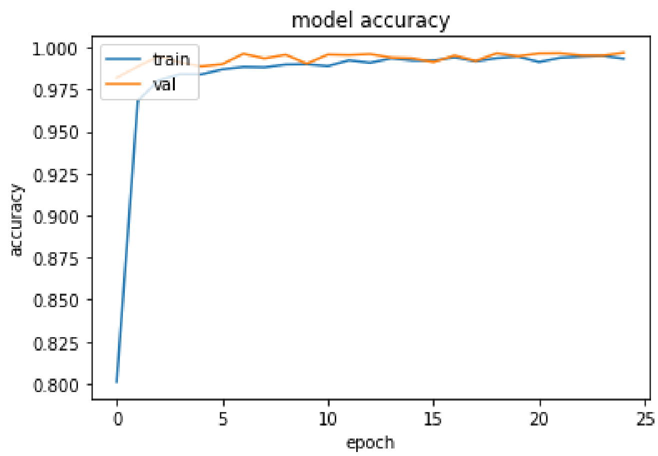

Fig. 3: Training accuracy and validation accuracy of proposed modified EfficientNetB6 architecture

G. Augmentation

For achieving better performance, the deep learning models need more examples while training. Often the input data are not sufficient. There are a couple of main problems behind acquiring adequate data. Firstly, the real-world datasets are imbalanced and can reduce the recognition accuracy as the model will not have enough samples to learn from. Secondly, the test images can be flipped, rotated, scaled, or noisy. For example, if we rotate, scale, flip a cat picture, it will still be a cat picture. Augmentation is the process of solving this dilemma which generates computational images from the input samples by utilizing processes like rotating, scaling, zooming, shearing, flipping, adding noise, etc. [23].

IV. Experimental Analysis

A. Preprocessing

Convolutional neural networks don't need excessive feature engineering as these models are capable of extracting important features by themselves. However, resizing the images was necessary as both InceptionV3 and EfficientNetB6 take images that have an input size of 224x224x3. Moreover, for better training of the parameters, augmentation was utilized via the augmentor library. Rotation function with maximum left and right rotation of 3 and 40% probability of rotation was utilized while applying augmentation. For the zoom random function, 90% area was utilized with a probability of 20%, and the random distortion function utilized grid height and width of 4 with 40% probability having a magnitude of 4. The augmentation process ended with 400 pictures per group, 40,800 images for 102 groups in total.

TABLE II: Comparison between our proposed works and notable previous works

Short Description of Methods/Process	Overall Accuracy
Convolutional Neural Network [8]	65.40%
PCA on SIFT Features [9]	83.90%
EL+YcbCr Technique [10]	78.00%
K-mean Reduction + CNN [11]	85.78%
Augmentation + CNN [12]	86.90%
CNN on SIFT Features [13]	89.70%
Convolutional Neural Network [14]	91.80%
VGG-16 [15]	94.38%
ResNet-50 [15]	91.13%
MobileNet [15]	92.07%
DenseNet-121 [15]	89.50%
NASNetMobile [15]	87.77%
CNN on Classifical Features [16]	90.10%
Proposed Modified InceptionV3	**99.65%**
Proposed Modified EfficientNetB6	**99.72%**
Proposed Softmax-averaging	**99.85%**

B. Experimental Design

The learning rate was fixed to 0.0001 for both modified InceptionV3 and modified EfficientNetB6 architectures. The models were run for 25 epochs each. The batch size was fixed to 24 which was the highest value that our machine could handle. 'Adam' optimizer [24] was utilized for optimization. Momentum and RMSprop optimizations were skipped as 'Adam' optimizer utilizes both aspects of them. A categorical cross-entropy function was practiced for the loss function. To bypass overfitting, dropouts were utilized and tuned for both models. ReLU activation was utilized for all convolution layers.

C. Result Analysis

The process started by splitting the dataset into 80:20 ratios where 80% of the data were kept for training and 20% data were kept as the test set. After applying augmentation on the training data, the train data was split in 80:20 as well where 80% data were kept in the training set and the rest of the 20% data were kept in the validation set. The difference between the validation set and the test set is that the validation set is a part of the training phase as the goal of training is to minimize the training and validation loss at the same time.

On the other hand, the test set is totally independent and has no impact on training. After splitting the dataset, modified InceptionV3 and modified EfficientNetB6 were applied. The reason for modifying the InceptionV3 and EfficientNetB6 models by adding additional layers is to channel the most important features only to the decision making output layer. Modified InceptionV3 achieved an overall accuracy of 99.65% and modified EfficientNetB6 achieved 99.72% accuracy. Both of them suppressed the previous studies by a fine margin. But to boost up the overall performance even more, we utilized the softmax-averaging technique and obtained 99.85% overall accuracy. Fig. 2 shows the training and validation accuracy while the training phase for modified InceptionV3 architecture. On the other hand, Fig. 3 shows the training and validation accuracy while training phase for modified EfficientNetB6 architecture. TABLE I illustrates the class-wise precision, recall, and accuracy where TABLE II represents the comparison in terms of overall accuracy among our proposed models and previous works. It can be noticed that our proposed models have suppressed the previous works by a noteworthy boundary.

V. Conclusion

In this research, we started with a benchmark dataset, Caltech-101 with 102 groups with a variety of objects i.e., people, faces, furniture, structures, animals, etc. which made the recognition more difficult. We utilized modified InceptionV3, modified EfficientNetB6, and softmax-averaging technique for the recognition obtaining 99.65%, 99.72%, and 99.85% overall accuracy respectively. Augmentation was also introduced in this work. After comparing the results, it turned out that our models have suppressed the previous works by a distinct margin. Moreover, InceptionV3 and EfficientNetB6 utilize 23.8 million and 43.2 million parameters - 67 million parameters in total. On the other hand, the previous best result obtained via VGG-16 utilized 138.3 million parameters which made our process more cost-effective. However, 67 million is still a huge number and implementation in mobile devices is still challenging. Designing a low-cost CNN model for object recognition is the most promising future work in this domain. Investigating with other datasets and on real-life pictures may reveal some possible future works as well.

References

[1] S. Dasiopoulou, V. Mezaris, I. Kompatsiaris, V.-K. Papastathis, and M. G. Strintzis, "Knowledge-assisted semantic video object detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 10, pp. 1210–1224, 2005.

[2] J. Wu, A. Osuntogun, T. Choudhury, M. Philipose, and J. M. Rehg, "A scalable approach to activity recognition based on object use," in *2007 IEEE 11th international conference on computer vision*. IEEE, 2007, pp. 1–8.

[3] L. Guan, Y. He, and S.-Y. Kung, *Multimedia image and video processing*. CRC press, 2012.

[4] S. Ahlawat and A. Choudhary, "Hybrid cnn-svm classifier for handwritten digit recognition," *Procedia Computer Science*, vol. 167, pp. 2554–2560, 2020.

[5] X. Wu, C. Sun, T. Zou, L. Li, L. Wang, and H. Liu, "Svm-based image partitioning for vision recognition of agv guide paths under complex illumination conditions," *Robotics and Computer-Integrated Manufacturing*, vol. 61, p. 101856, 2020.

[6] N. Wang, Y. Wang, and M. J. Er, "Review on deep learning techniques for marine object recognition: Architectures and algorithms," *Control Engineering Practice*, p. 104458, 2020.

[7] M. Z. Alom, M. Hasan, C. Yakopcic, T. M. Taha, and V. K. Asari, "Improved inception-residual convolutional neural network for object recognition," *Neural Computing and Applications*, vol. 32, no. 1, pp. 279–293, 2020.

[8] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 609–616.

[9] J. Song, G. Yoon, H. Cho, and S. M. Yoon, "Structure preserving dimensionality reduction for visual object recognition," *Multimedia Tools and Applications*, vol. 77, no. 18, pp. 23 529–23 545, 2018.

[10] M. A. Khan, T. Akram, M. Sharif, M. Awais, K. Javed, H. Ali, and T. Saba, "Ccdf: Automatic system for segmentation and recognition of fruit crops diseases based on correlation coefficient and deep cnn features," *Computers and electronics in agriculture*, vol. 155, pp. 220–236, 2018.

[11] Y. Pan, Y. Xia, Y. Song, and W. Cai, "Locality constrained encoding of frequency and spatial information for image classification," *Multimedia Tools and Applications*, vol. 77, no. 19, pp. 24 891–24 907, 2018.

[12] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation strategies from data," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 113–123.

[13] M. Rashid, M. A. Khan, M. Sharif, M. Raza, M. M. Sarfraz, and F. Afza, "Object detection and classification: a joint selection and fusion strategy of deep convolutional neural network and sift point features," *Multimedia Tools and Applications*, vol. 78, no. 12, pp. 15 751–15 777, 2019.

[14] Y. Sawada, Y. Sato, T. Nakada, S. Yamaguchi, K. Ujimoto, and N. Hayashi, "Improvement in classification performance based on target vector modification for all-transfer deep learning," *Applied Sciences*, vol. 9, no. 1, p. 128, 2019.

[15] S. Basha, S. K. Vinakota, S. R. Dubey, V. Pulabaigari, and S. Mukherjee, "Autofcl: Automatically tuning fully connected layers for transfer learning," *arXiv preprint arXiv:2001.11951*, 2020.

[16] N. Hussain, M. A. Khan, M. Sharif, S. A. Khan, A. A. Albesher, T. Saba, and A. Armaghan, "A deep neural network and classical features based scheme for objects recognition: an application for machine inspection," *Multimed Tools Appl. https://doi. org/10.1007/s11042-020-08852-3*, 2020.

[17] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 4, pp. 594–611, 2006.

[18] M. Valueva, N. Nagornov, P. Lyakhov, G. Valuev, and N. Chervyakov, "Application of the residue number system to reduce hardware costs of the convolutional neural network implementation," *Mathematics and Computers in Simulation*, 2020.

[19] J. West, D. Ventura, and S. Warnick, "Spring research presentation: A theoretical foundation for inductive transfer," *Brigham Young University, College of Physical and Mathematical Sciences*, vol. 1, no. 08, 2007.

[20] J. Tang, *Intelligent Mobile Projects with TensorFlow: Build 10+ Artificial Intelligence Apps Using TensorFlow Mobile and Lite for IOS, Android, and Raspberry Pi*. Packt Publishing Ltd, 2018.

[21] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 6105–6114.

[22] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le, "Mnasnet: Platform-aware neural architecture search for mobile," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2820–2828.

[23] M. D. Bloice, C. Stocker, and A. Holzinger, "Augmentor: an image augmentation library for machine learning," *arXiv preprint arXiv:1708.04680*, 2017.

[24] Z. Zhang, "Improved adam optimizer for deep neural networks," in *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*. IEEE, 2018, pp. 1–2.