Image Classification using Convolutional Neural Network on Ciphar-10 and MNIST datasets

Abdulnour Dualeh

ec211178@quml.ac.uk

***Abstract*—Deep Learning algorithms are supposed to emulate the human cerebral cortex's function. These algorithms are deep neural network representations, with many hidden layers. With 2D pictures as input, convolutional neural networks can train massive datasets with millions of parameters, and then filter them to create desired outputs. On picture recognition and detection datasets, CNN models are created in this article. The algorithm's performance is assessed using MNIST and CIFAR-10 datasets.**

***Index Terms*—CIFAR-10, MNIST, VGG16, ResNet34, AlexNet**

I. Introduction

Detection and recognition of images is another well-known problem in machine learning. Detecting an item or recognizing an image from a digital picture or video is extremely difficult. For example, facial recognition, biometric systems, self-driving cars, emotion detection, picture restoration and robotics are just some of the numerous uses of Image Recognition [1].

There has been a lot of improvement in computer vision thanks to Deep Learning algorithms. An artificial neural network with several hidden layers, called a "deep learning" implementation, is used to imitate the cerebral cortex's operations. There are various degrees of abstraction provided by the layers of a deep neural network. In contrast to shallow networks, this does not have the capability to extract or work on many features. One of the most effective deep learning algorithms, convolutional neural networks (CNNs) can cope with millions of parameters while reducing the processing cost by convolutional of a 2D picture and creating output volumes [1].

Handwritten digits are used in the MNIST dataset, which measures the accuracy of a classification method. In addition to OCR, signature verification, text interpretation and modification, and many more uses, handwritten digit recognition offers numerous other benefits. A recent achievement in the realm of handwritten digit identification is an image classification and recognition problem. Classifying things into ten categories and then detecting them in test sets is the CIFAR10 object detection dataset. Image detection methods may be implemented using this collection of photos.

The MNIST dataset and the CIFAR-10 dataset are used to test the performance of Convolutional neural network models in image recognition and object identification, respectively. Model implementation is described, and the correctness of the results is reviewed. On the CIFAR-10 dataset, a single CPU unit is employed to train the model and real-time data augmentation is used. The datasets are also protected against Overfitting by using Dropout.

Image classification and computer vision are two of the most widely used applications of modern computation. The RGB (Red, Green, and Blue) channels are used to interpret and understand the image. It signifies that the image has A columns, B rows, and 3 color channels if the image size is A B 3. For a fully - connected network, the first hidden layer would need a large number of weights because of the high pixels per inch in most photos [2].

It's in this situation that a Convolutional neural network really shines, as each of its neurons is only linked to a small portion of the layer below it. Pixel-by-pixel comparisons with a source images are used by computers to extract the relevant data from these warped images. Sections of the image are called features, and the computer uses them to compare the two photos. It is much easier to identify and identify photos by comparing just a portion of them rather than the complete image.

Figure 1 Diagram of CNN and its components as well as mathematical operators are depicted below. Using a filter of a certain size moved over the picture, a feature map is created using the CNN's Convolution layer as the first step in extracting the image's features [2].

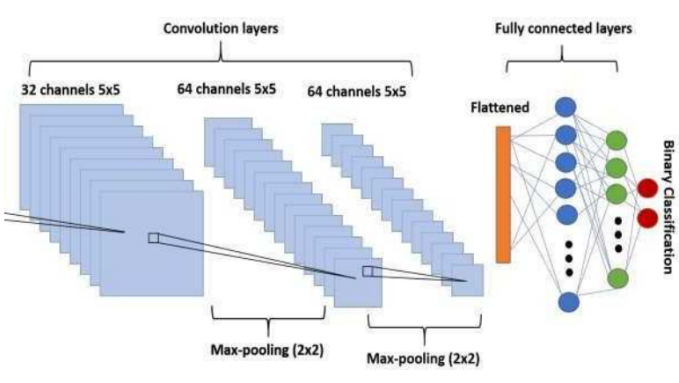

Fig. 1. Diagram of CNN and its components as well as mathematical operators [2]

The result of shifting the filter is a matrix of positive and negative pixel values, which is the feature map. Rectified linear activation function (ReLU) accepts the preceding layer's output as input and, if positive, provides the previous layer's

output as an output. Nonetheless, if the input pixel value is discovered to be negative, then ReLU's output will be zero. Following this, the Pooling layer attempts to minimize the size of the feature map. As a result, computing time is cut in half, resulting in a speedier overall procedure. A variety of pooling techniques can be used, including Max Pooling and Average Pooling. Using a combination of a Convolution layer, ReLU layer, and Pooling layer results in a further reduction in input size. The last layer of any neural network is the fully connected layer of the CNN. As a result, CNN's piece is not a one-off. Activation functions (logistic or Softmax) are applied to the input vector to the fully connected layer in order to determine the probability. The existence of the needed feature in the picture would indicate a successful image detection if the probability was higher.

II. Related Work

The paper attempts to apply AlexNet, LeNet-5, and VGG Net on the CIFAR-10 dataset and describe their characteristics and performance. Multiple training techniques have been tested: AlexNet, LeNet5, VGG Net. These artificial neural algorithms are recognized for their competition accuracy. Although not intended for CIFAR-10 training, they would be used to test the model's adaptability. So they'll tweak it and test it on CIFAR-10 [3].

Similarly on the CIFAR-10 dataset, authors compare the performance of several classifiers and design an ensemble of classifiers to improve performance. Convolutional Neural Network (CNN) and K-Nearest Neighbors (KNN) are mutually exclusive on CIFAR-10, this will result in a greater accuracies achieved when they are combined. They use Principal Component Analysis(PCA) to decrease overfitting and combine it with a CNN to improve its accuracy. this method increases the best CNN model from 93.33 to 94.03 [4].

The Cifar-10 dataset is used to test their deep learning model. Various regularization approaches and function optimization methods like Adam and RMS are utilized to improve picture classification accuracy [5].

Another author proposed the Discrete Cosine S-Transform to recognize handwritten English digits (DCST). The experiments used the freely available MNIST handwritten digit database. The DCST characteristics and an ANN classifier are used to classify handwritten digits. They are taken from the MNIST handwritten isolated digit database standard pictures. The database contains 70000 samples, 60000 for training and 10000 for testing. To reduce computational cost, they reduced the MNIST dataset picture size from 28x28 to 20x20 by removing unsought border pixels up to width four. A back propagation neural network was also used to classify digits. This work has 98.8 percent success rate for MNIST dataset [6].

Using several convolutions, ReLU, and pooling layers, we constructed an efficient model. Which has 98.45 percent accuracy on MNIST data. The proposed model is also tested on a comparable random picture data set with significant accuracy results [7].

Another Author work proposes SVM technique for MNIST dataset using fringe and inverse fringe as SVM feature. MNIST data collection has 60000 training examples and 10000 test instances. In our experiments, we found that using the fringe distance map as a feature increased the system's accuracy to 99.99 percent on trained data and 97.14 percent on test data, while using the inverse fringe distance map increased the system's accuracy to 99.92 percent on trained data and 97.72 percent on test data [8].

A CNN-based model with great accuracy was built. This sparked my interest in handwritten character categorization using machine learning. The Kannada-MNIST (K-MNIST) dataset was released in 2019. We chose this dataset for our handwritten character categorization study. A CNN model was created for picture classification problems because to its unique design. This study examines the establishment and experimentation of the CNN model. We compared our model's performance to various machine learning approaches such as Logistic Regression and Support Vector Machines. The model outperformed all baselines and obtained 98.77 percent accuracy on the K-MNIST testing set (0-9). Thus, we determined that CNN models can classify handwritten characters well [9].

III. Convolutional Neural Networks Architectures

A. AlexNet

Now we can discuss our CNN's entire architecture. As shown in Figure 2, the net has eight weighted layers, five of which are convolutional and three of which are completely linked. From the last fully-connected layer, a 1000-way softmax generates a distribution across 1000 class labels. It optimizes multinominal logistic regression maximizing the log-probability average across training instances under the prediction distribution [10]. The second, fourth, and fifth convolutional layers' kernels are linked solely to those preceding layer maps on the same GPU (see Figure). The third's kernels Convolutional layer connects to second layer's kernel maps.

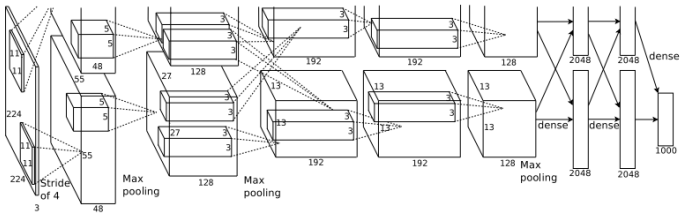

Fig. 2. Architecture of AlexNet

The completely linked layers are connected to the previous levels. Normalization layers 2nd convolutional layer follows. Max-pooling layers, as in Section 3.4, then the response-normalization and fifth convolutional layers. ReLU. Every convolutional and fully-connected layer's output is nonlinear. Filtering the 224x224x3 input picture with 11x11x3 size of 96 kernels. Using a stride of 4 pixels (between nearby receptive

field centers in a kernel map). The second convolutional layer filters the first convolutional layer's output (response-normalized and pooled) with 256 kernels of size 5x5x48.The third, fourth, and fifth convolutional layers are directly coupled [10].

layering or normalizing The third convolutional layer contains 384 kernels of 3x3x256 linked to the second convolutional layer's (normalized) outputs. The fourth convolutional layer contains 384 kernels of size 3x3x256, while the fifth has 3x3x192 size of 256 kernels Each layer has 4096 neurons. An image of the CNN's structure, clearly demonstrating the division of duties between the two graphics cards. Using one GPU to run layer-parts at the top and another to run layer-parts at the bottom, Only some layers of the GPUs are interconnected. A 150,528 dimensional input is sent into the network. A network's remaining layers include a total of 253,440–186,624–64,896–43,264–64,896 neurons 4096–4096–1000.

B. VGG16

Our ConvNets are fed a fixed-size 224x224x RGB picture during training. The only preprocessing we do is remove each pixel from the mean RGB value calculated on the training set. We utilize filters with a very narrow receptive field: 3x3 (which is the lowest size to capture the notions of left/right, up/down, and so on) to send the picture through a stack of convolutional (conv.) layers. center). We also use 1 1 convolution filters in one of the combinations, which can be visualized as a transformation of the input channels that is linear (followed by non-linearity) [11]. The stride of convolution is set to 1 pixel; conv. layer input spatial padding is such that spatial resolution is retained After convolution; the padding is 1 pixel for each of the three convolution layers. The process of spatial pooling is carried out by Some of the conv. layers are followed by five max-pooling layers (not all of the conv. layers are followed). maximum-pooling). Max-pooling is done with stride 2 across a 2x2 pixel frame. Following that is a stack of convolutional layers (of varying depth in different topologies). There are three Fully-Connected (FC) layers: the first two each contain 4096 channels, while the third does 1000-channel operations.

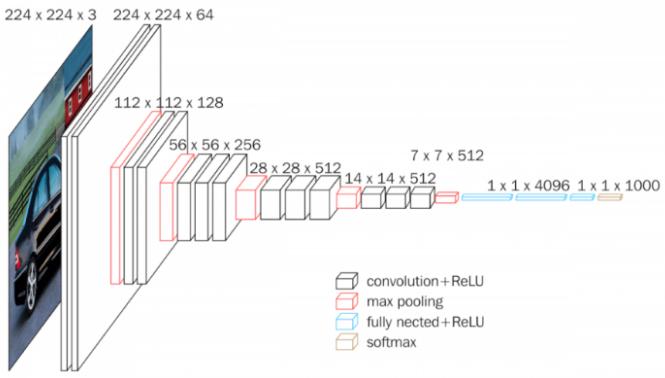

Fig. 3. Architecture of VGG16

There are 1000 channels in the ILSVRC categorization system (one for each class). The topmost layer is the layer of soft-max In all networks, the completely linked levels are configured the same way [11]. The rectification non-linearity is present in all buried levels. Except for one, none of our networks have Local Response Normalization does not enhance performance on the ILSVRC dataset but increases memory use and calculation time. The LRN layer's settings are those for the LRN layer when appropriate.

C. ResNet34

The concept of VGG nets [12] inspired our plain baselines. The convolutional layers typically contain 33 filters and follow two basic design rules: the layers will have the same number of kernel for the same output feature map size; and also the number of kernels is twice as much if the feature map size is half to maintain the time complexity per layer. We use convolutional layers with a stride of 2 to do direct down sampling. We also use a global average pooling layer and a 1000-way Dense layer with softmax activation to complete the network.

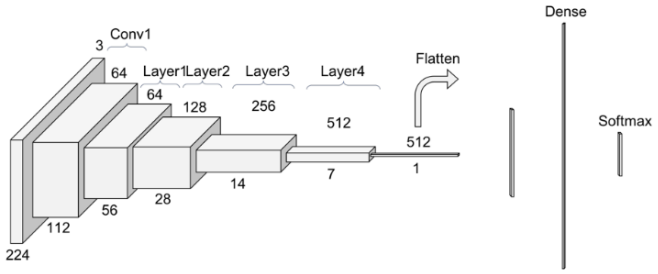

Fig. 4. Architecture of ResNet34

Network of residuals We inject shortcut connections into the aforementioned simple network to transform it into its residual counterpart. When the input and output dimensions are the same, the identity shortcut can be applied directly. We investigate two possibilities as the dimensions expand. The shortcut still conducts identity mapping, but with more entries that are 0, to account for the increased dimensions. This option does not add any new parameters; When the shortcuts traverse feature maps of two sizes in both choices, they are done with a stride of 2 [12].

IV. Experiments

Here we will discuss and explain how the experiments were carried out.

A. Datasets

For this experiment we have used two different datasets. Each dataset has its own purpose and characteristics.

B. CIFAR-10

There are total of 60000 images in the CIFAR-10 dataset [13]. There is total 10 classes which includes airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. So, there are total to 6000 images in each class.. The resolution of the images is 32x32 pixels. Each of the 10000 photos in the dataset is separated into five batches of training and one batch of test. Each class has a total of 1000 photos in the test batch. There are certain batches that have more photos through one class than another, but this is not consistent. There are a total of 5000 photos from each class in the training batches. Class members can't mix and match from any of these subgroups. Automobiles and trucks do not share any classifications. In the automotive context, cars, SUVs, and other vehicles of this type are included. Only large trucks are considered "trucks." Both exclude pickup vehicles [13].

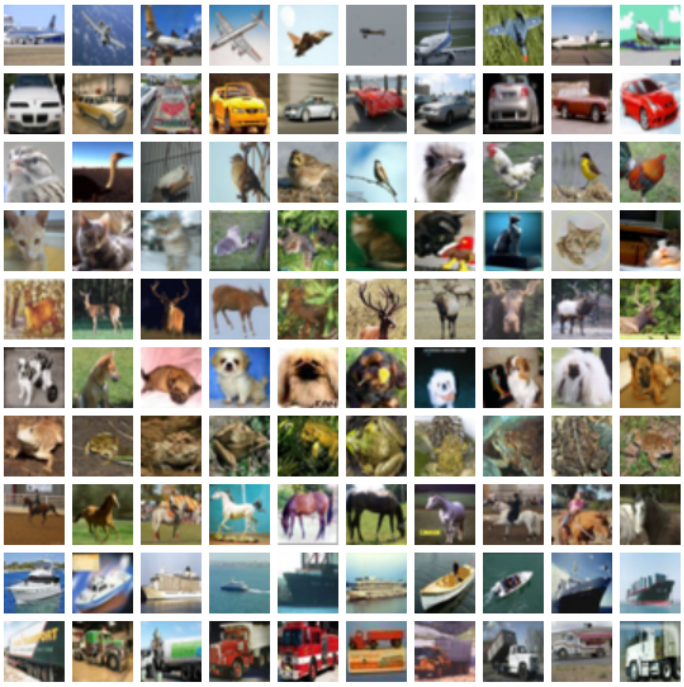

Fig. 5. Samples from CIFAR-10 Dataset

C. MNIST

Over a million numbers have been scribbled into what is known as the MNIST database (Modified National Institute of Standards and Technology) [14]. Training and testing sets total 60,000 samples and 10,000 examples, respectively. Special Database 3 (numbers written by workers of the US Census Bureau) and Special Database 1 (digits produced by high school students) include monochrome photographs of handwritten digits. A fixed-size picture has been used to center the digits after they have been size-normalized. The 20x20 pixel container was created by resizing the original NIST white and black photos to fit. The normalization algorithm's anti-aliasing approach yields photos with a variety of grey tones. After calculating the load of the pixels, then interpreting the picture so that this point was located in the middle of a 28x28 field, images were centered [14].

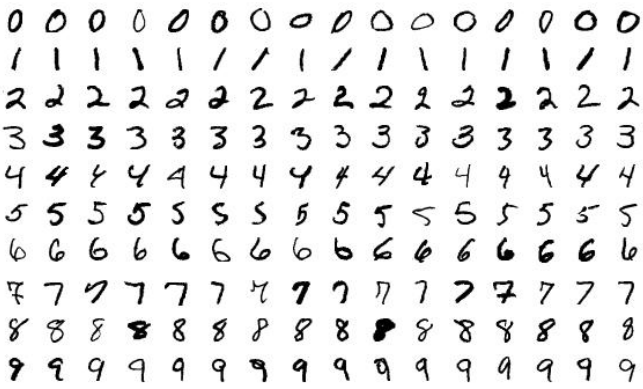

Fig. 6. MNIST Dataset

V. Testing Results

A. MNIST training

1) RESNET: The Resnet model uses a Learning rate of 0.01 and a batch size of 4 for the experiment on the MNIST dataset. The model is trained on the MNIST dataset for 10 epochs and received a final training accuracy of 99.82% and a test accuracy of 99.45%. The model generalizes well and therefore would give accurate predictions. For each of the 10

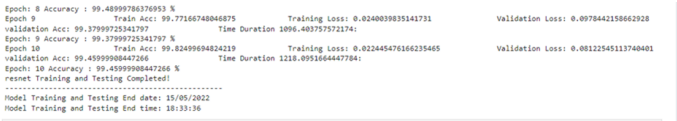

Fig. 7. RESNET Results on MNIST dataset

classes in the MNIST dataset, these are the model predictions.

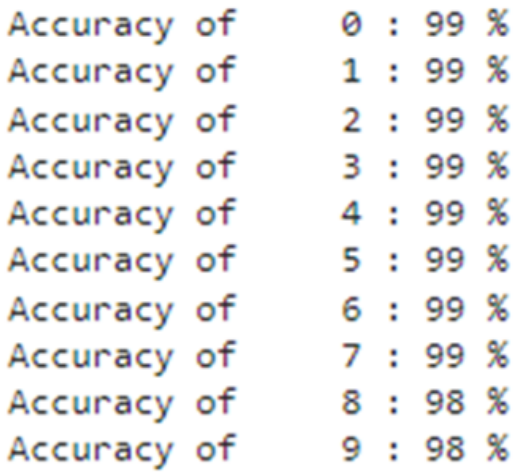

Fig. 8. RESNET Results on MNIST dataset

We can see that the resnet model is accurately predicting the class of images.

2) AlexNet: For this model a batch size of 64 and learning rate of 0.01 was used. The model achieved a training accuracy

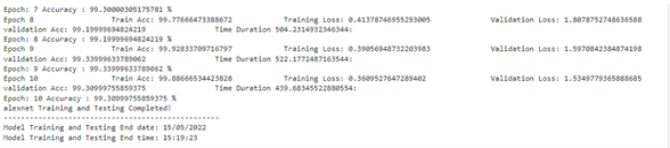

Fig. 9. RESNET Results on MNIST dataset

of 99.8% and a validation accuracy of 99.3%. This model is generalising well and has a balanced fit.

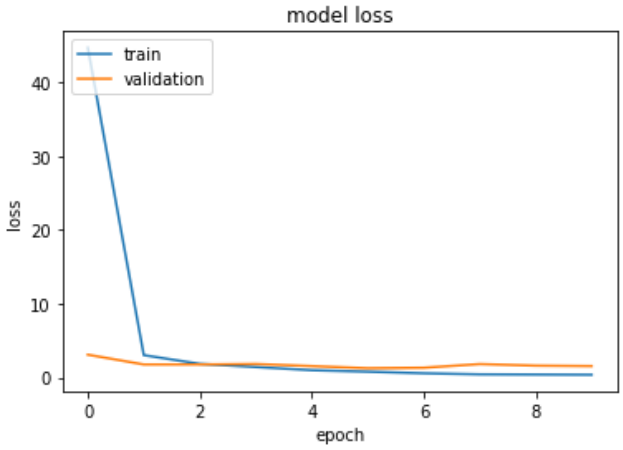

Fig. 10. RESNET Results on MNIST dataset

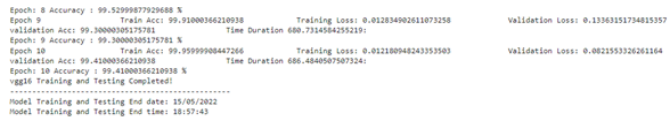

Fig. 11. RESNET Results on MNIST dataset

3) VGG-16: The model achieved 99.9% training accuracy and 99.4% validation accuracy The model had a balanced fit and generalised well.

B. CIFAR-10 Training

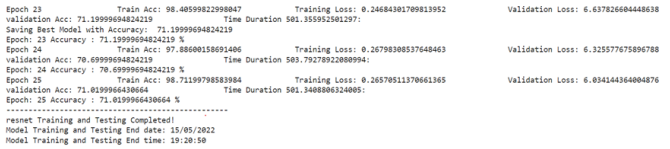

Fig. 12. RESNET Results on MNIST dataset

1) ResNet: The Model achieved a training accuracy of 98.4% and a validation accuracy of 71.1%. The model suffers from overfitting clearly from the training and validation accuracy, as the model fails to generalize and has a low bias, which makes the model vulnerable to the noise in any image.

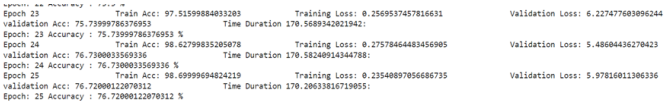

Fig. 13. RESNET Results on MNIST dataset

2) AlexNet: The model achieved a training accuracy of 98.7% and a validation accuracy of 76.7%. Despite the model suffering from overfitting, still performs better than the Resnet model in generalizing.

3) VGG-16:

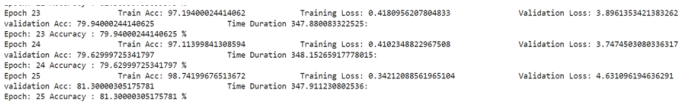

Fig. 14. RESNET Results on MNIST dataset

4) AlexNet:

Fig. 15. RESNET Results on MNIST dataset

5) AlexNet: The model achieved 98.7% training accuracy and 81.3% validation accuracy, despite the model suffering from overfitting, as we can see from the loss curve, and low bias, the model actually performs better than the previous two models(Alexnet and Resnet) models in generalizing.

6) Comparison: From Table I, we can see that VGG-16 performed the best across the datasets, and have the lowest training loss. It is alos clear to see that Resnet performed better than Alexnet on the MNIST dataset, but in the CIFAR-10 dataset it had an overall accuracy of 71.09% therefore, it performed the worst.

Model	Dataset	Number of Epochs	Train Loss	Test Loss	Train accuracy%	Test accuracy%
VGG-16	MNIST	10	0.012	0.082	99.96	99.48
Alexnet	MNIST	10	0.363	1.53	99.88	99.30
Resnet	MNIST	10	0.022	0.081	99.82	99.45
VGG-16	CIFAR-10	25	0.342	4.631	98.74	81.30
Alexnet	CIFAR-10	25	0.235	5.978	98.69	76.72
Resnet	CIFAR-10	25	0.265	6.034	98.71	71.09

TABLE I
FINAL RESULTS OF MODEL TRAINING AND TESTING

VI. Conclusion

These models performed well on the Mnist dataset as it's very low features for each class and can be easily represented and so therefore less training was required for the model to generalise well. however on the CIFAR-10 dataset the models suffered from overfitting, and there're some model improvements that can be made to improve results for both the model and the data.

Model improvement techniques include Local response Normalization (LRN), l2 regularization and dropout to prevent overfitting and increase the generalization ability of the model. Deepening the number of network layers and residual network technology can increase the fitting ability of the model by deepening the number of model layers and solving the problem of gradient attenuation. These improved methods stack step by step, step by step, making the fitting ability and generalization ability of the network stronger and stronger, and finally obtain higher classification accuracy.

References

[1] R. Chauhan, K. K. Ghanshala, and R. C. Joshi, "Convolutional Neural Network (CNN) for Image Detection and Recognition," ICSCCC 2018 - 1st International Conference on Secure Cyber Computing and Communications, pp. 278–282, Jul. 2018, doi: 10.1109/ICSCCC.2018.8703316.

[2] S. Mascarenhas and M. Agarwal, "A comparison between VGG16, VGG19 and ResNet50 architecture frameworks for Image Classification," Proceedings of IEEE International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications, CENTCON 2021, pp. 96–99, 2021, doi: 10.1109/CENTCON52345.2021.9687944.

[3] X. Zhang, "The AlexNet, LeNet-5 and VGG NET applied to CIFAR-10," Proceedings - 2021 2nd International Conference on Big Data and Artificial Intelligence and Software Engineering, ICBASE 2021, pp. 414–419, 2021, doi: 10.1109/ICBASE53849.2021.00083.

[4] Y. Abouelnaga, O. S. Ali, H. Rady, and M. Moustafa, "CIFAR-10: KNN-Based Ensemble of Classifiers," Proceedings - 2016 International Conference on Computational Science and Computational Intelligence, CSCI 2016, pp. 1192–1195, Mar. 2017, doi: 10.1109/CSCI.2016.0225.

[5] R. Doon, T. Kumar Rawat, and S. Gautam, "Cifar-10 classification using deep convolutional neural network," 1st International Conference on Data Science and Analytics, PuneCon 2018 - Proceedings, Nov. 2018, doi: 10.1109/PUNECON.2018.8745428.

[6] R. K. Mohapatra, B. Majhi, and S. K. Jena, "Classification performance analysis of MNIST Dataset utilizing a Multi-resolution Technique," 2015 International Conference on Computing, Communication and Security, ICCCS 2015, Jan. 2016, doi: 10.1109/CCCS.2015.7374136.

[7] A. Garg, Di. Gupta, S. Saxena, and P. P. Sahadev, "Validation of Random Dataset Using an Efficient CNN Model Trained on MNIST Handwritten Dataset," 2019 6th International Conference on Signal Processing and Integrated Networks, SPIN 2019, pp. 602–606, May 2019, doi: 10.1109/SPIN.2019.8711703.

[8] A. Patel and T. v. Kalyani, "Support Vector Machine with Inverse Fringe as Feature for MNIST Dataset," Proceedings - 6th International Advanced Computing Conference, IACC 2016, pp. 123–126, Aug. 2016, doi: 10.1109/IACC.2016.32.

[9] E. X. Gu, "Convolutional Neural Network Based Kannada-MNIST Classification," 2021 IEEE International Conference on Consumer Electronics and Computer Engineering, ICCECE 2021, pp. 180–185, Jan. 2021, doi: 10.1109/ICCECE51280.2021.9342474.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," Commun ACM, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[11] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, Sep. 2014, doi: 10.48550/arxiv.1409.1556.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2016-December, pp. 770–778, Dec. 2015, doi: 10.48550/arxiv.1512.03385.

[13] "CIFAR-10 and CIFAR-100 datasets." https://www.cs.toronto.edu/ kriz/cifar.html (accessed May 14, 2022).

[14] "MNIST handwritten digit database, Yann LeCun, Corinna Cortes and Chris Burges." http://yann.lecun.com/exdb/mnist/ (accessed May 14, 2022).

[15] "PyTorch: Transfer Learning and Image Classification - PyImageSearch." https://pyimagesearch.com/2021/10/11/pytorch-transfer-learning-and-image-classification/ (accessed May 14, 2022).