Journal of Theoretical and Applied Information Technology
15th December 2022. Vol.100. No 23

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

PERFORMANCE ANALYSIS AND EVALUATION OF IMAGE CLASSIFICATION MODELS USING MACHINE LEARNING

[1]MOHAMED NOUR, [2]RASHA M. AL-MAKHLASAWY, [3]MAYADA KHAIRY
Electronics Research Institute, Cairo, Egypt
E-mail: [1]mnour@eri.sci.eg, [2]rashamostafa@eri.sci.eg, [3]mayada@eri.sci.eg

ABSTRACT

This work presents an image classification process using machine and deep learning. The machine learning has feature extraction and classification modules. It can extract certain features of images but unable to select differentiating features from the training set of data. Deep learning can find naturally the relevant features for the adopted applications. The convolutional neural network (CNN) is one of the common deep learning approaches. CNN has an input layer, hidden layers, and an output layer. An image is constructed as a matrix of pixels where the pixel values are given to the input layer supported with weights and biases. The hidden layers are convolutional, pooling and/or fully connected layers. The output layer is a fully connected layer to classify the image to which class it belongs to. Moreover, a set of hyper-parameters are analyzed and investigated. The parameters play an important role in the performance of the image classification process. A set of experiments are operated to see the effect of every hyper parameter. The parameters include; but not limited to; the number of hidden layers, the number of epochs, filter size, number of filters, batch size, learning rate, optimization method, and others.

Moreover, a useful supervised machine learning approach is adopted to classify the images. The number of selected features has a vital role on the performance of the support vector machine (SVM). Both the CNN and SVM are operated and tested using two big datasets. The first dataset; CIFAR-10; has ten classes and 60,000 images where the second one; MNIST; has ten classes and 70,000 images. The performance of both deep learning and SVM approaches are compared. Some measurable criteria are considered such as accuracy, learning time, prediction time, and others. The classification accuracy using CNN outperforms that accuracy value for the SVM. The performance of CNN using the MNIST dataset is better than the CNN using the CIFAR-10 dataset. This means that the dataset size, nature, and characterization play an important role in the performance of machine and deep learning approaches. The learning time and prediction time for CNN approach are greater than those corresponding values of SVM. The obtained results in this work are better than some of the related efforts published in the literatures by others using the same machine learning approaches and the same datasets.

Keywords: *Image Classification, Machine Learning, Deep Learning, Image Datasets, and Performance Evaluation.*

1. INTRODUCTION AND RELATED WORK

Image classification aims to process digital images using computer algorithms. The important steps of any image processing task include: importing the image using an acquisition tool, analyzing the acquired image, and producing results based on the image analysis. Several operations can be done on an input image either to get an enhanced image or to extract some useful information. Image processing involves several important themes such as pattern recognition, image editing, image segmentation, image restoration, multi-scale image analysis, feature extraction, image classification, and others.

Image classification is an important task to categorize and label groups of pixels or vectors within an image based on specific rules. Supervised image classification aims at selecting training data within the image and assessing it to one of the predefined categories or classes. Deep learning is an important type of machine learning which can utilize a layered structure of several algorithms expressed as an artificial neural network (ANN). ANN can be simulated with the help of the biological neural network of the human brain. In most cases, deep learning approaches outperform those classical ones specially when using big amount of data [1]. Images can be classified to their

relevant predefined categories using machine and deep learning approaches. Image classification is important for several applications such as healthcare systems, medical images, industrial applications, video games, and others [2], [3] and [4]. Moreover, several research efforts were presented for handling machine learning, deep learning, feature extraction, feature selection, image classification, and others. Examples of such efforts for image classification include; but not limited to; the following:-

[5] discussed some image classification methods based on artificial intelligence (AI) for diagnosing the skin cancer. Some AI solutions were developed; specially deep learning algorithms; to distinguish malignant skin lesions from benign lesions in different image modalities such as clinical, histopathology, and dermascopic images. The authors highlighted some challenges and future opportunities to improve the AI solutions to support dermatologists and enhance their ability to diagnose the skin cancer.

[6] mentioned that deep learning can be used to improve the performance of image processing. Deep learning is promising for different application areas such as: agriculture, space agencies, medical field, forensics, and others. The authors presented the architectures of some deep learning approaches and also the applications of deep learning in image detection, image segmentation, and image classification. The authors highlighted the benefits and weaknesses of deep learning tools that are used for image processing.

[4] reviewed CNN deep learning models and compared their differences and similarities. Target detection and object segmentation algorithms were presented. The adopted deep learning approach was used to solve some problems in computer vision, multi-objective classification and relevant fields in industry and academia. Beside the innovation of deep learning algorithms, the construction of large-scale datasets and development tools were important for image classification.

[7] proposed a method for image classification and object recognition. The proposed method is based on amalgamating both the SVM and CNN. The Alex-Net was pre-trained for the large-scale object image dataset. The SVM was used as trainable classifier. The feature vectors were passed to the SVM from Alex-Net. The STL-10 dataset was used as object images where the number of classes was ten. The STL-10 object images were trained by the SVM with data augmentation. Some augmentation methods were applied such as rotation, skewing and elastic distortion. The experimental results for applying the proposed image classification and image augmentation were effective and promising.

[8] mentioned that the deep learning model has a powerful learning ability which integrates the feature extraction and classification process to complete the image classification test. This played an important role for improving the image classification accuracy. The authors proposed an image classification algorithm based on the stacked sparse coding depth learning model-optimized kernel function non-negative sparse representation. The experimental results of the proposed method presented higher accuracy values compared with some related works. Also, the proposed method was used to improve the image classification accuracy in complex problems.

[9] applied some machine learning algorithms for image classification based on the bag of features approach. The bag of features aims to find vector representation of input images that categorized images into a finite set of classes. The authors presented a comparative study between different feature extraction methods and classification algorithms. The authors made guessing of the best machine learning technique to recognize the stop sign images.

[10] discussed some machine learning approaches to make pattern recognition. The approaches are multilayer perception, SVM, CNN and others. The authors mentioned that the process of identification of the symbol and different numbers are based on the methods of machine learning. From the experimental results, the Bayesian neural networks were promising for classification.

The organization of this work is as follows: Section 2 gives a brief overview of the adopted test-bed datasets for image classification. Section 3 discusses the convolution neural network (CNN) deep learning model. Section 3 also presents the main building blocks of the CNN deep learning model. Section 4 presents the implementation work and experimental results for applying the CNN and SVM machine learning approaches on the test-bed datasets. The effect of each key parameter of the deep learning model is also discussed and investigated. All the obtained results are compared and discussed in Section 5. Finally, Section 6 concludes the whole work.

2. THE ADOPTED DATASETS FOR IMAGE CLASSIFICATION

In this research work, two common image datasets are applied as test-beds. The datasets are CIFAR-10 and MNIST. The CIFAR-10 dataset consists of 60,000 colored images distributed in 10 classes with 6,000 per class. The input image is of size 32x32x3 and each image subtracts its own three channel (R/G/B) mean value to speed-up the conversion of deep CNN model. The size of the whole dataset is 170 MB. The classes of CIFAR-10 are: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. During the classification process 50,000 images and 10,000 images are dedicated for training and testing respectively [11], [12], and [3].

The second dataset is called MNIST and it is concerned with handwritten data. The number of records of MNIST is 70,000 images in 10 classes where 60,000 images are dedicated for training while the remaining 10,000 images are used for testing. All images are in grayscale with 10 classes. The size of the whole MNIST dataset is 30 MB [13], [14].

3. AN ADOPTED DEEP LEARNING MODEL

Machine learning plays an important role for a lot of applications. Machine learning aims at automatically learn to make predictions as well as classification based on the previous observations. Image classification can utilize the benefits from machine learning in general and from deep learning in particular. Deep learning uses multiple layers to represent the abstraction of data. The deep learning approaches are promising as they can amalgamate the feature extraction and image classification.

Deep learning models based on convolution neural network (CNN) presented a great attention from several researchers in image classification. CNN can learn directly from the image data so there is no need to make manual feature extraction. The advantages of CNN; include but not limited to; parameter sharing, equivalent representation and sparse interactions [15], [16], [17], [8], and [18]. CNN can be used to extract and compress the features of an image and obtain the higher level ones. CNN is a promising approach because it is designed to support non-linear and sophisticated data manipulation for effective learning.

3.1 Why Deep Learning Based on Convolution Neural Network?

Convolution neural network (CNN) is effective in feature representation performance in addition to its capability to achieve complex image classification. The analysis of CNN is adopted in this research work as it has a lot of advantages compared to some other deep learning methods. Using CNN, the input image matches well with the topological CNN structure. Extraction of features can occur simultaneously with pattern classification and generation in the training process. CNN can be effectively used to recognize 2D images in terms of shifting, scaling and some sort of distortion. CNN can avoid explicit feature extraction and learns implicitly from data training. CNN can learn in parallel as the neurons in the same feature mapping plane share the weight. CNN uses multi-layer convolution and trains with a fully connected layer. In multilayer convolution, the higher the layer the more global the learnt features are. CNN; in most cases; is more flexible and cost efficient than some other related methods. Moreover, real time deep learning using CNN is important for real time image classification [19], [13], and [20].

3.2 The Main Architectural Units of CNN Deep Learning Model

CNN can be used for cognitive tasks and image processing in general and image classification in particular. The CNN architecture has an input layer, number of hidden layers and an output layer. Some layers are convolved using mathematical models to fetch and prepare results to the succeeding layers. The CNN model has an input layer, convolutional and pooling layers followed by one or more fully connected layers, and an output layer. When an image is provided, the CNN can extract the features from the image using multiple pairs of convolutional and pooling layers and the image can be classified into a class using fully connected layers. The main architectural outlines of CNN are briefly shown in Figure 1 at the end of the paper.

In the CNN architectural model, there are parameters that play an important role in learning time, prediction time and the classification accuracy. This includes; but not limited to; the number of layers, number of filters, filter size, stride size, learning rate, number of epochs, batch size, pooling window, and others. A brief description of CNN main constructs is presented in the following sub-sections [13], [20], [21], [22], and [23].

3.2.1 The Input Layer

In the input layer, the network takes all information needed. An image is comprised of pixels where each pixel has three color elements red, green, and blue. Each element ranges from 0 (i.e. no color) to 255 (full saturation). A color image is considered a three layered matrix of pixels where each layer is a two-dimensional matrix representing red, green, or blue pixel values. A gray scale image is stored as 2D matrix.

3.2.2 Convolution Layers

The convolution layers are important and play a vital role in the CNN performance. The term 'Convolution' merges and/or combines two functions to form a third one. A convolution layer sometimes called filter or kernel and it can be operated on the input data to generate what is called a feature map. The CNN has multiple convolution layers where the first layer detects the low-level features of the input image such as color, gradient orientation, and edges. The next layers are focused to detect the middle level features such as the image shapes. The last layers are interested in detecting an object. Figure 2 presents a multiplication operation which is done between a filter matrix of size 3x3 and a part of the input image matrix of size 3x3. The elements of the resulting matrix are summed as a destination pixel. Such pixel is considered as an output value on the feature map. The same convolution operation is repeated several times. i.e. the filter slides over the input matrix, repeats the dot product multiplication with every remaining combination of 3x3 sized areas, and then completes the feature map. [13], [23], and [24]. By performing the element-wise multiplications and sum, the first feature is produced in the feature map. A new feature map is generated each time by sliding the filter a certain number of positions specified by the stride size. Figure 2 presents an example of convolution using a kernel of size 3x3. The element-wise multiplications between the input data and the filter are done and summed up to produce the output. In this example the stride is 1 and the filter slides by one position to the right or down after completing each convolution operation [13].

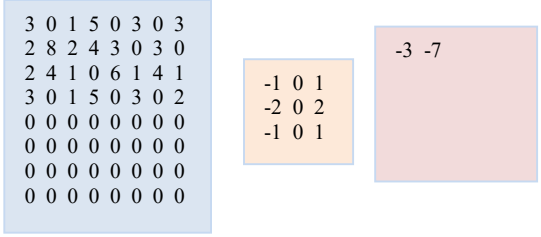


Figure 2: The Filter Slides Over the Input and Performs its Output on the New Layer [Hung-Dao, 2020]

3.2.3 Pooling Layers

The inputs to the pooling layers are fed from the output of the previous convolutional layers. The pooling layers are concerned with reducing the dimensionality of the feature maps specifically the height and width preserving the depth. It is important to use the pooling layers to reduce the dimensionality and the risk of overfitting. i.e. the convolutional power to process the data is reduced while extracting the dominant features in the feature maps. The pooling layer has no parameters, and it down-samples the result from the previous layer which is known as data compression. The down-sampling process in this data is adopting the max pooling.

As the max pooling is preferred than the average pooling (as mentioned in several literature), the max pooling is adopted in this work. The max pooling can output the maximum value of the elements in the part of the image covered by the filter. As examples, the max pooling of size 2x2 with depth 1 and stride 2 are briefly mentioned in Figure 3. The max pooling, window size, stride, and number of pooling layers play a vital role as they produce better time and accuracy. The max pooling is applied for dimensionality reduction via down-sampling.

It is easy to say that the operation in the pooling layer regarding the image processing is considered as transforming a high resolution image into a low resolution image. After handling both the convolution and pooling layers, the number of parameters in the CNN model can be reduced [13], [20], and [23].

Moreover, the activation function indicates whether a neuron is activated or not. There are several activation functions such as ReLu function, Sigmoid function, and Tanh function and others. In this research work, the ReLu activation function is adopted as it is the most effective function for the CNN as mentioned in several literatures [23], [20], [12], [11]. The ReLu function is a non-linear function used in this work and can be briefly described as follows:-

$$F(x) = \max(0, x) \quad (1)$$

The range of formula 1 is $(0, +\infty)$, and its derivative is

$$f'(x) = 0 \text{ if } x < 0, 1 \text{ if } x > 0, \text{ and undefined if } x = 0 \quad (2)$$

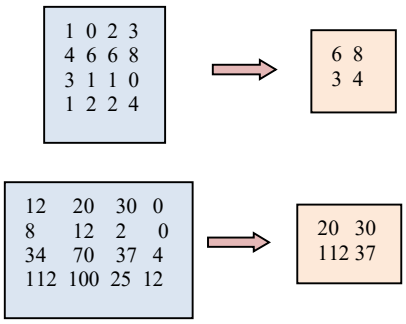


Figure 3: Two Max Pooling are Applied for Dimensionality Reduction Via Down-Sampling

3.2.4 Fully Connected Layers

The feature maps which were processed through the convolution and pooling layers are converted to a single dimensional vector and fed to the fully connected layer. i.i a fully connected layer in CNN has several neurons and it can be represented in a column vector. All neurons are connected via weights where the convolutional layer is converting the weights into a huge matrix. In such matrix, most entries are considered 0 except in designated regions and many regions share the same weight [20]. In another way, each neuron in the first fully connected layer computes the weighted sum of the features provided to itself. The fully connected layer computes the weighted sum of the output signals provided as the input to itself. This process is repeated through the fully connected layers [13]. If the layers of convolution, pooling and activation function map the original data into the feature space in the hidden layer, the fully connected layer maps the learnt distributed-feature-representation into the sample label space. Generally speaking, the convolution and pooling layers seem to be like the feature engineering, while the full connection seems to be like the feature weighting [20].

3.2.5 Output Layer

Assuming that different features with their different weights are given, the convolution and fully connected layers can find the most correlated features to a particular class. The output layer is focused to give the probabilities where the input image belongs to different predefined classes. In fact, this is based on the detected features [13].

Moreover, each fully connected layer is passed through an activation function (ReLu in our case) then the output is passed through the softmax function. For image multi-class classification, the softmax function is a common approach to compute the probabilities [23]. The output of softmax function is an N-dimensional vector, where N is the number of classes. The CNN model selects one of such classes [13], [20], and [23].

3.3 Image Classification Using Supervised Machine Learning

There are several supervised machine learning approaches; one of them is the support vector machines (SVM). It is known that SVM can build a hyper plane or a set of hyper planes in a high dimensional space for classification. SVM as a classifier can solve the pattern recognition problems with two classes. The best decision of hyper plane occurs when it separates a set of positive examples from a set of negative examples with maximum margin. The multi-class SVM can classify the set of images by finding the best hyper-plane that separates all images of one class from those of the other classes. The margin is considered the maximum width to the hyper-plane that has no interior data points. The performance of the SVM classifier depends on the hyper-plane selection and kernel parameter. For more details about SVM, readers can refer to [30], [31], and [32]. The CNN deep learning model and SVM machine learning approach are applied to classify the images of the adopted datasets: CIFAR-10 and MNIST. The performance of each classifier is evaluated.

4. IMPLEMENTATION WORK AND EXPERIMENTAL RESULTS

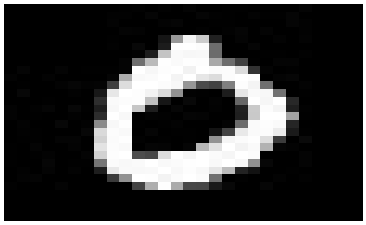
The CNN and SVM approaches are implemented, applied and tested by presenting a set of experiments. All experiments are operated using a laptop supported by Windows-10 operating systems, installed RAM 16.0 GB, and Processor Intel® Core ™ i5-3210M CPU@ 2.5GHZ processing speed. The adopted machine learning models are implemented and run using Matlab-R2019a. As mentioned above, two datasets are used to evaluate the performance of CNN and SVM learning approaches. A set of experiments are operated and tested to monitor the effectiveness of the hyper-parameters. The number of combinations for the CNN architectural models is about one-hundred and sixty-five which are operated and applied on the two datasets. The hyper-parameters are: learning rate, batch size, number of epochs, filter size, stride size, and others. The hyper-parameters are briefly defined as shown below [25], [24], [34], [35], and [36].

Number of Epochs: The number of epochs is a hyper-parameter concerned with defining the number of times that the learning model works through the entire training dataset [25].
Batch size: The batch size is a hyper-parameter that defines the number of samples to work through before updating the internal model parameters. A sample contains inputs that are fed into the model and an output that is used to compare to the prediction and calculate an error [35].
The number of convolution layers: The number of convolution layers has a significant effect on the performance. The bad choice of that number may cause over-fitting or under-fitting problems. Over-fitting and under-fitting may occur when the number of hidden layers is large or small compared respectively with the problem complexity [23].
Learning Rate: The learning rate is a hyper-parameter that controls how much to change the model in response to the estimated error each time the model weights are updated. Too small values of learning rates may result in a long training process that could get stuck, whereas a value too large may result in learning to be too fast or an unstable training process. The amount that the weights are updated during training is referred to as the step size or the "learning rate" [35] and [36].
Number of Filters: In the CNN model, multiple filters are used for one input. The resulting feature maps are joined together for the final output of one convolution layer.
Filter size: The filter size is defined as a continuous variable, which is optimized by minimizing the training loss. The unit for filter size is pixel [23].
Fully connected layers: In CNN model, the feature maps processed through the convolution and pooling layers are flattened. A fully connected layer has many neurons and can be represented in a column vector. The neurons in this layer are connected via weights and the fully connected layer can be transformed into a convolution layer and vice versa. The fully connected layer works as a classifier in the entire CNN model [20] and [23].
Stride: The stride is a component of CNN tuned for the compression of images and video data. Stride is a parameter of the neural network's filter that modifies the amount of movement over the image or video. For example, if a stride is set to 1, the filter will move one pixel at a time. The filter size affects the encoded output volume, so stride is often set to a whole integer, rather than a fraction or decimal [23] and [53].
Moreover, the hyper-parameters play an important role in the classifiers' performance. To see the impact of each parameter, only one parameter is changed while all the other parameters are kept fixed. i.e. a group of experiments are done for different values assigned to parameter#1 (for example) while all the other parameters are assigned fixed values. Similarly, a set of experiments are also done for different values of parameter#2 while fixing the assigned values for the other parameters. That process is repeated for all the adopted hyper-parameters. The same concept is also done for the two adopted datasets: Table 1 shows the setting values for the adopted hyper parameter. Moreover, Figures 4 and 5 present respectively one image from each of CIFAR-10 and MNIST.

4.1 Experimental Results for Classifying CIFAR-10 Images Dataset

As mentioned before, this dataset has ten classes; each image is composed of 32X32X3 pixels with three RGB colors. 50,000 images were used for training and 10,000 for testing. A set of experiments are operated for classifying the images dataset considering the different hyper-parameters mentioned above. The classification accuracy, learning time, and prediction time are registered for different values of the hyper-parameters as shown in Figure 6.

(a)



(b)

Figure 4: (a): An Image from CIFAR-10 (b): An Image from MNIST [Shonqing Gu, et. al., 2019]

4.2 Experimental Results for Classifying MINST Images Dataset

As mentioned before, the MNIST image dataset is concerned with handwritten digits. It has 60,000 samples for training and 10,000 samples for testing. Each grey-scale image is of size 28x28. The dataset consists of ten classes of images of digits from 0-9 [36]. A set of experiment are operated for classifying the images dataset considering the

different hyper-parameters mentioned above. The classification accuracy, learning time, and prediction time are also computed and illustrated for different values of hyper-parameters as shown in Figure 7.

4.3 Experimental Results for Classifying Images Using SVM

Several experiments are applied to classify the images of CIFAR-10 and MNIST using the SVM supervised machine learning approach. Feature selection plays a vital role in building any supervised classification approach. Feature selection mainly aims to reduce the number of attributes of the input images. The redundant and irrelevant attributes are better to be removed from the feature set as they can deteriorate the performance of the image classification process. In other words; feature selection is focused to consider only the most significant features of the input images. The most relevant features can achieve good classification accuracy and reduce the high dimensionality. Dimensionality reduction is important not only for improving the image-classification accuracy, but also for reducing the storage requirements. Some researchers consider the feature selection process as one of the preprocessing operation [37] and [38].

Moreover, there are several approaches for feature selection. The adopted approaches here aim to reduce the size of feature vector by transforming a higher dimensional feature space to a lower dimension. The adopted approaches are based on principal component analysis (PCA) and linear discriminate analysis (LDA). PCA is adopted to find the k-components that efficiently contain maximum variability of the original data. It transforms the original high dimensional data into lower dimensional components that are independent from each other. LDA is also used to transform the high dimensional data into a lower dimension [38]. The classification accuracy, learning time, and prediction time are illustrated in Figures 8 and 9 for the SVM classifier using respectively the CIFAR-10 and MNIST datasets.

5. DISCUSSION OF RESULTS

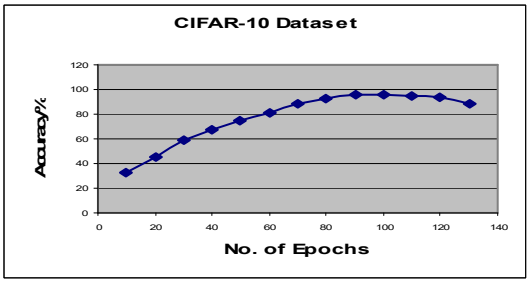
The CNN and SVM learning models were applied, tested and compared. The models were operated on the CIFAR-10 and MNIST datasets. CNN deep learning was adopted as it can learn directly from image data without any need of manual feature extraction. The SVM model was also chosen as it is one of the best supervised machine learning approaches for classification as mentioned in many published papers in the literatures. Several experiments were done for the learning approaches for each dataset. The experiments monitored the behavior of the key hyper-parameters which have significant effects on the performance. From the obtained results it was noticed that:-

The chosen activation function in the experiment was ReLU due to the advice of some research efforts published in the literatures. The unsaturated nonlinear activation functions; like ReLU; realize lower error rates than the saturated nonlinear activation functions such as Sigmoid. The adopted function is similar to the biological neurons which can improve the classification performance [20]. Moreover; there is some sort of trade-offs between the obtained accuracy and both the learning time and prediction time.

Figure 6a illustrates the accuracy values for different values of epochs. The best accuracy occurred when the number of epochs was 100. Also, the learning time and prediction time were increased by increasing the number of epochs as shown respectively in Figures 6c and 6d. The

Journal of Theoretical and Applied Information Technology
15th December 2022. Vol.100. No 23

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195



6a: Accuracy% Vs. No. of Epochs

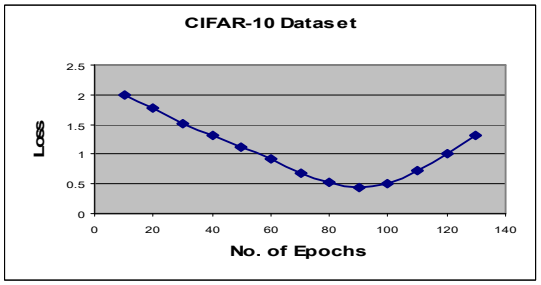


Figure 6b: Loss % Vs. No. of Epochs

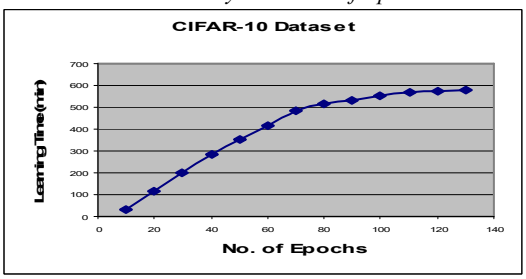


Figure 6c: Learning Time Vs. No. of Epochs

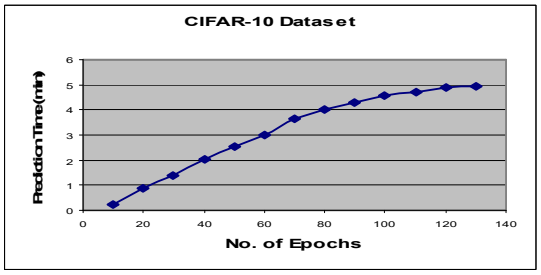


Figure 6d: Prediction Time Vs. No. of Epochs

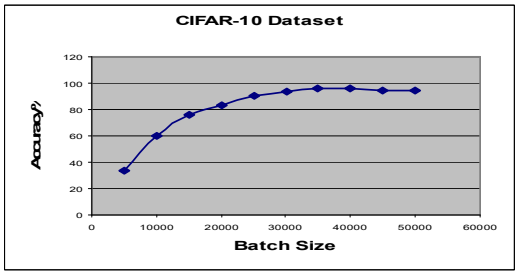


Figure 6e: Accuracy% Vs. Batch Size

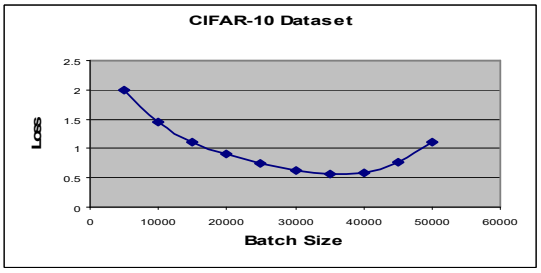


Figure 6f: Loss% Vs. Batch Size

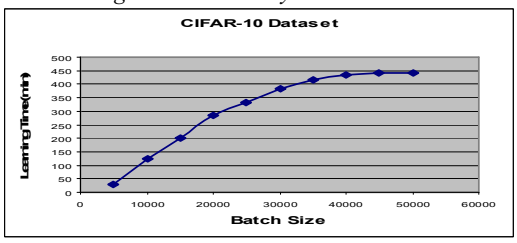


Figure 6g: Learning Time Vs. Batch Size

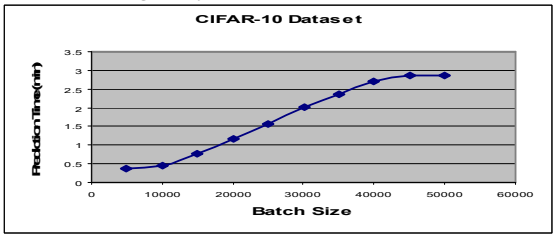


Figure 6h: Prediction Time Vs. Batch Size

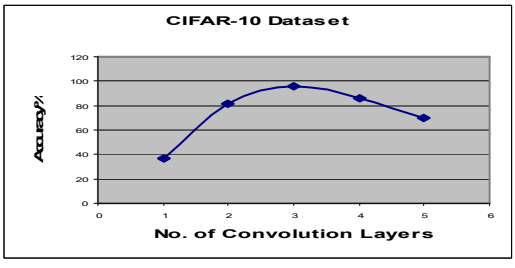


Figure 6i: Accuracy% Vs. # Convolution Layers

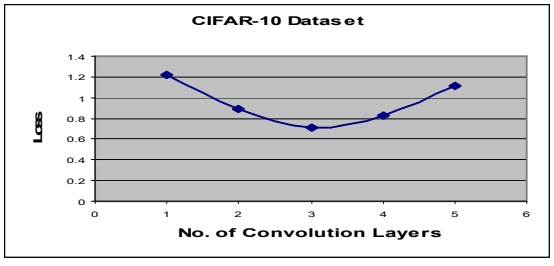


Figure 6j: Loss% Vs. # Convolution Layers

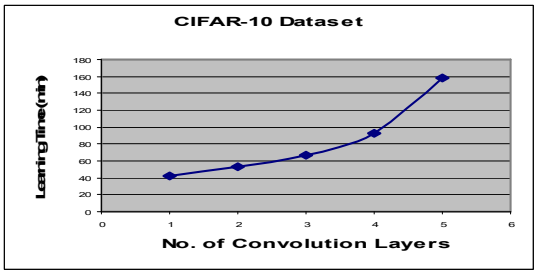


Figure 6k: Learning Time Vs. # Conv. Layers

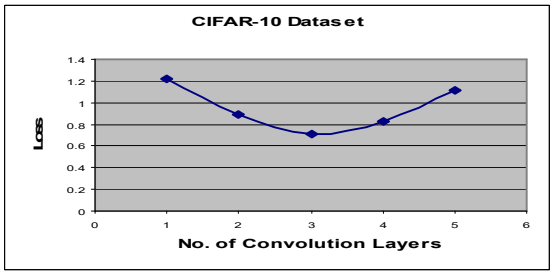


Figure 6l: Prediction Time Vs. # Conv. Layers

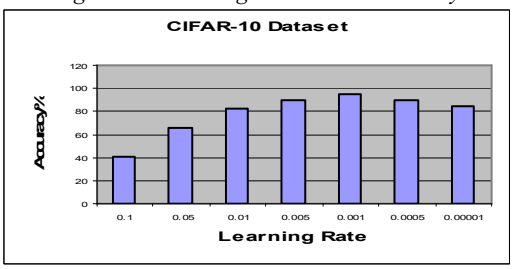


Figure 6m: Accuracy% Vs. Learning Rate

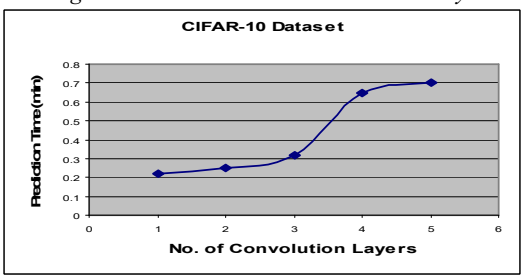


Figure 6n: Loss% Vs. Learning Rate

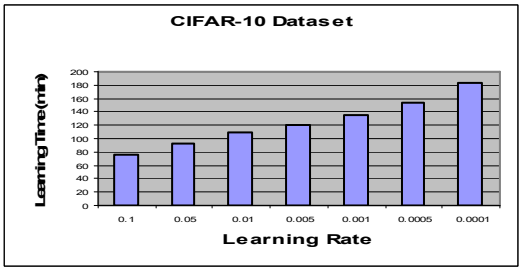


Figure 6o: Learning Time Vs. Learning Rate

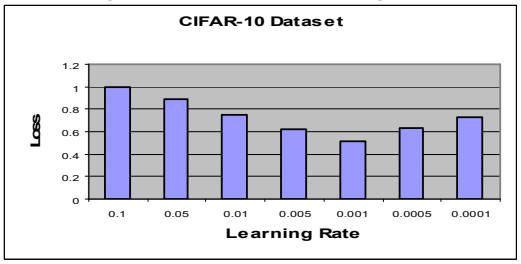


Figure 6p: Prediction Time Vs. Learning Rate

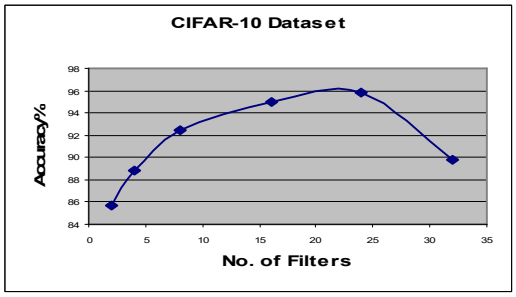


Figure 6q: Accuracy% Vs. No. of Filters

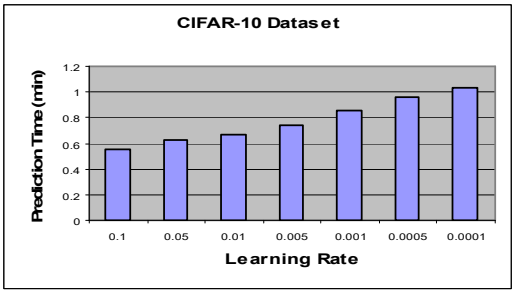


Figure 6r: Learning Time Vs. No. of Filters

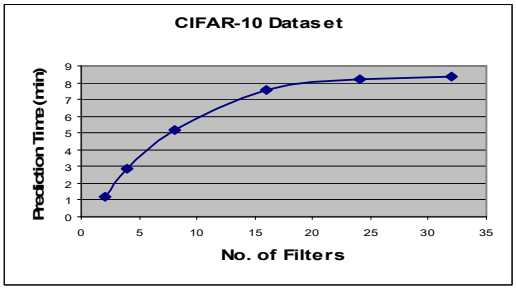


Figure 6s: Prediction Time Vs. No. of Filters

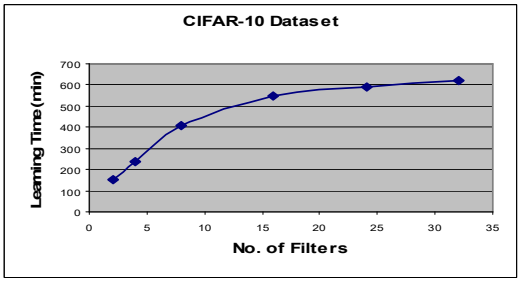


Figure 6t: Accuracy% Vs. Filter Size

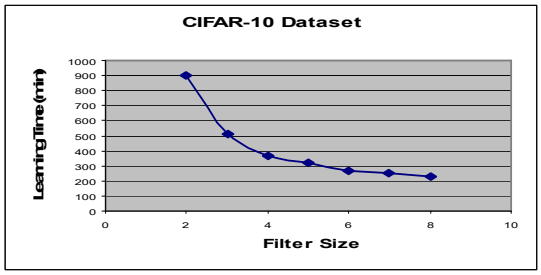


Figure 6u: Learning Time Vs. Filter Size

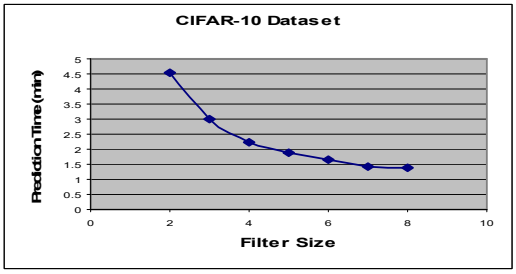


Figure 6v: Prediction Time Vs. Filter Size

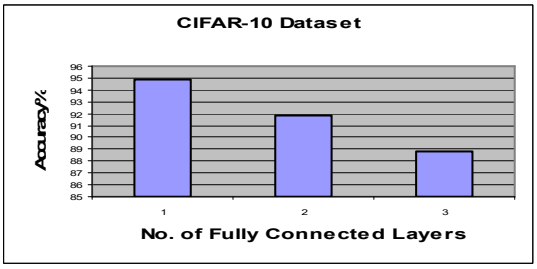


Figure 6w: Accuracy% Vs. # Fully Conn. Layers

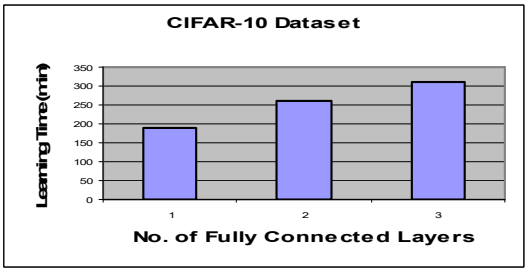


Figure 6x: Learning Time Vs. #Fully Conn. Layers

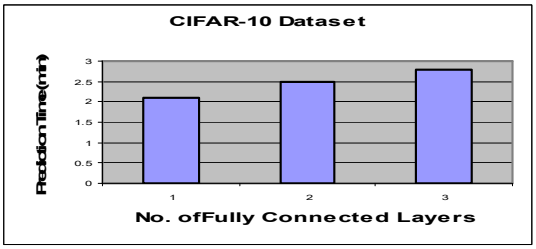


Figure 6y: Prediction Time Vs. # Fully Conn. Layers

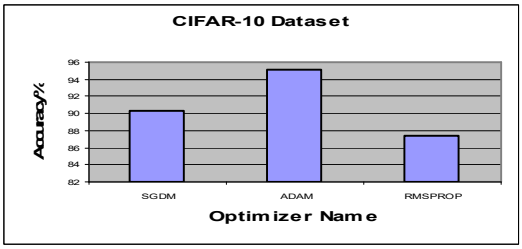


Figure 6z: Accuracy% for Optimizers

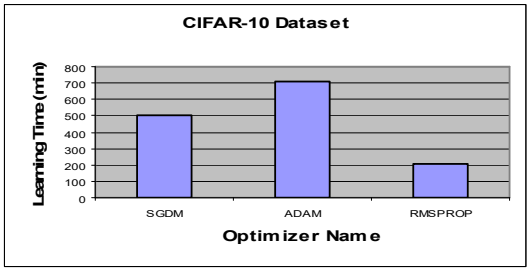


Figure 6aa: Learning Time for Optimizers

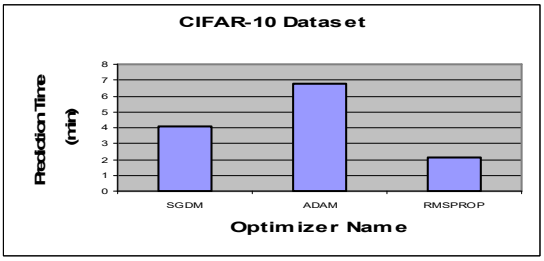


Figure 6ab: Prediction Time for Optimizers

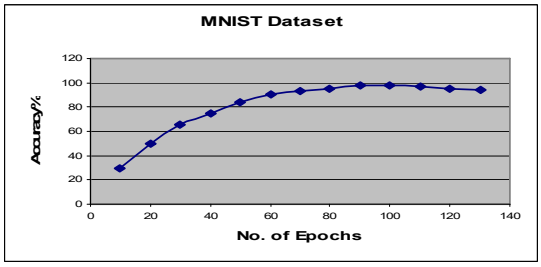


Figure 7a: Accuracy% Vs. No. of Epochs

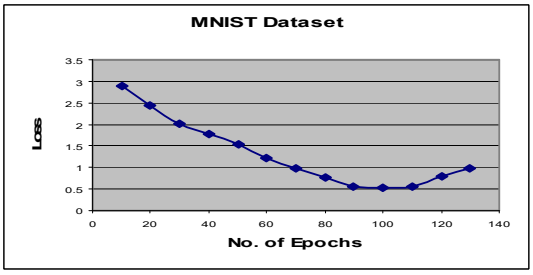


Figure 7b: Loss% Vs. No. of Epochs

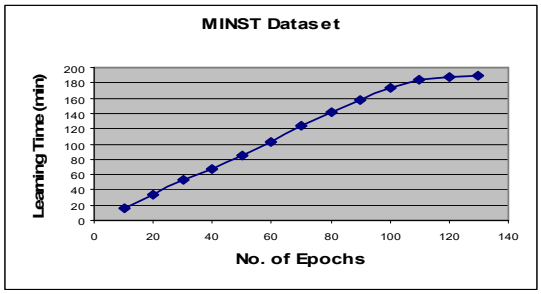


Figure 7c: Learning Time Vs. No. of Epochs

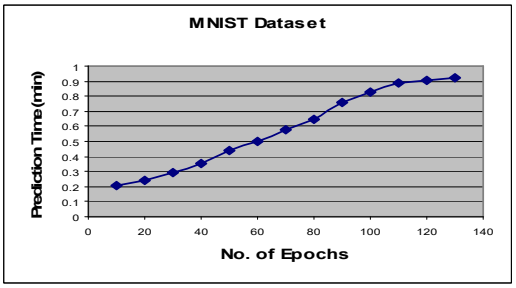


Figure 7d: Prediction Time Vs. No. of Epochs

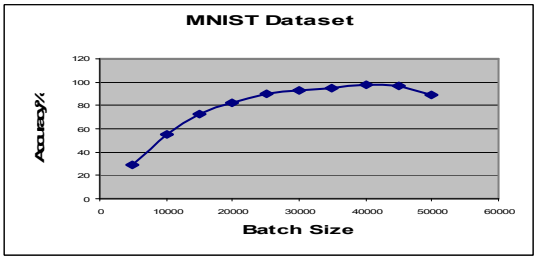


Figure 7e: Accuracy% Vs. Batch Size

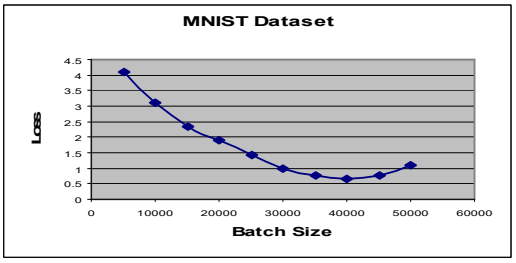


Figure 7f: Loss% Vs. Batch Size

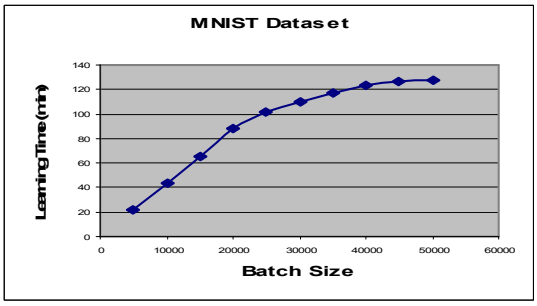


Figure 7g: Learning Time Vs. Batch Size

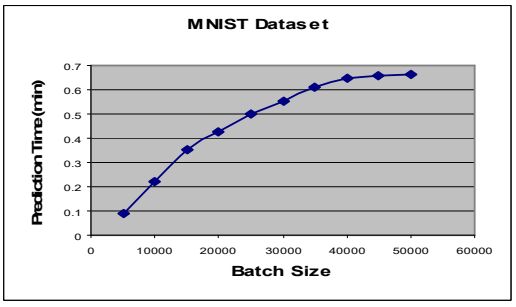
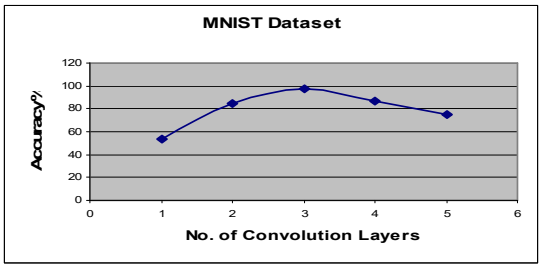


Figure 7h: Prediction Time Vs. Batch Size



7i: Accuracy% Vs. # Conv. Layers

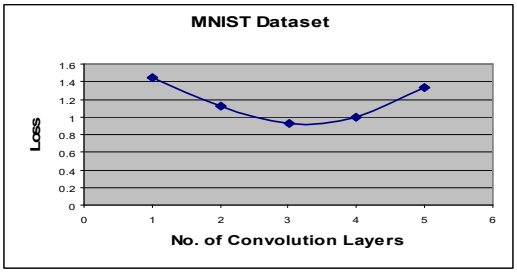


Figure 7j: Loss% Vs. # Conv. Layers

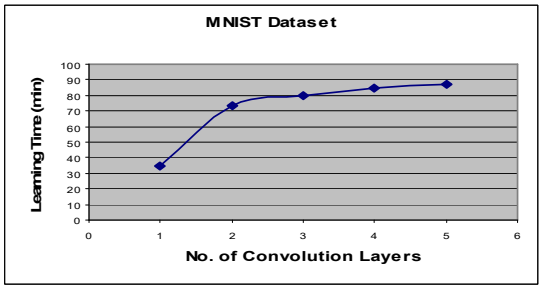


Figure 7k: Learning Time Vs. # Conv. Layers

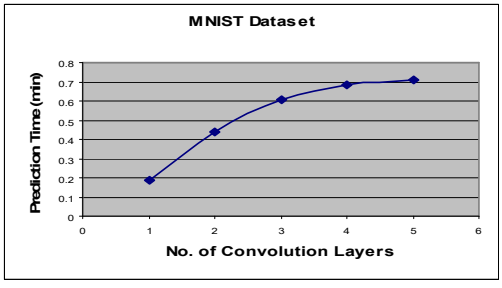


Figure 7l: Prediction Time Vs. # Conv. Layers

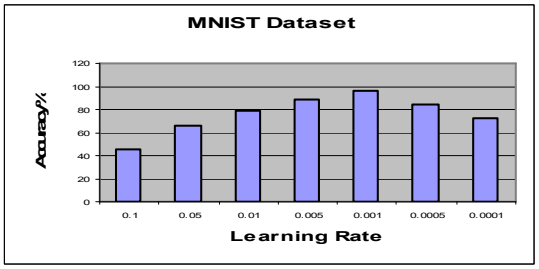


Figure 7m: Accuracy% Vs. Learning Rate

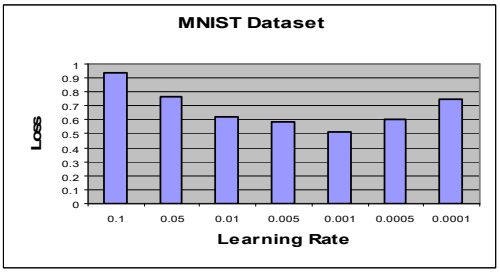


Figure 7n: Loss% Vs. Learning Rate

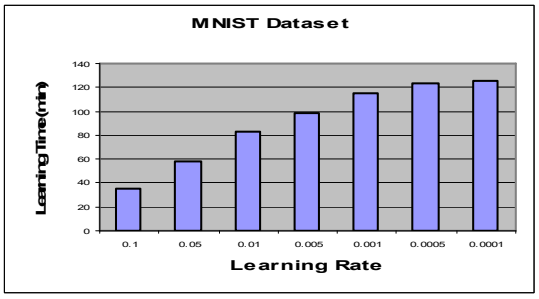


Figure 7o: Learning Time Vs. Learning Rate

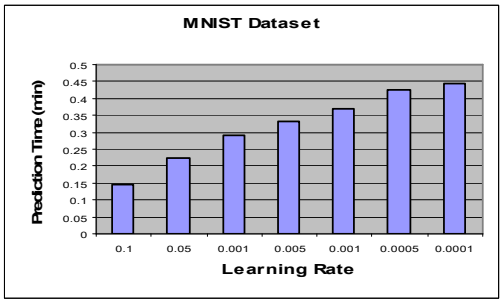


Figure 7p: Prediction Time Vs. Learning Rate

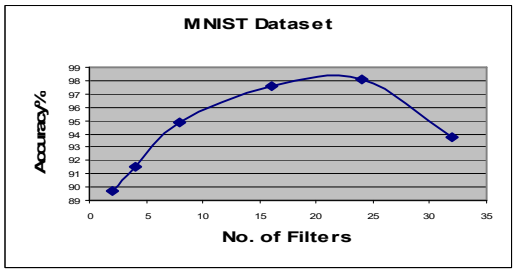


Figure 7q: Accuracy% Vs. No. of Features

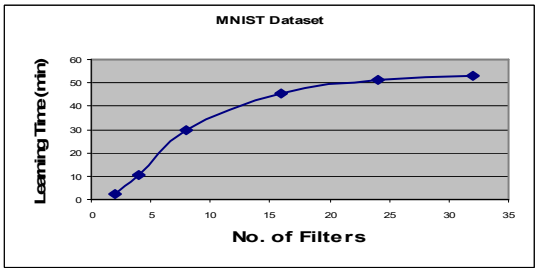


Figure 7r: Learning Time Vs. No. of Features

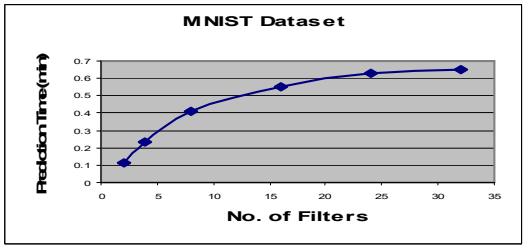


Figure 7s: Prediction Time Vs. No. of Features

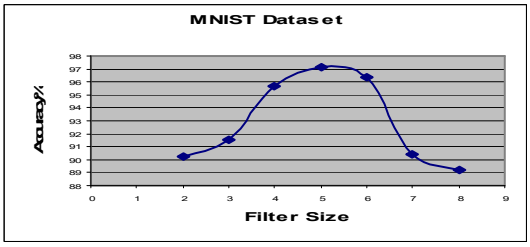


Figure 7t: Accuracy% Vs. Filter Size

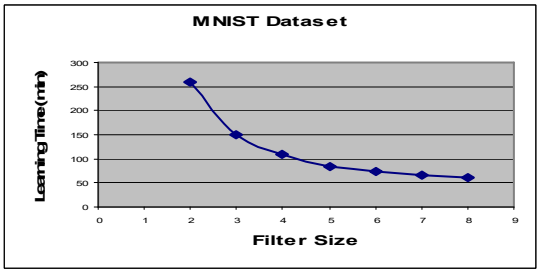


Figure 7u: Learning Time Vs. Filter Size

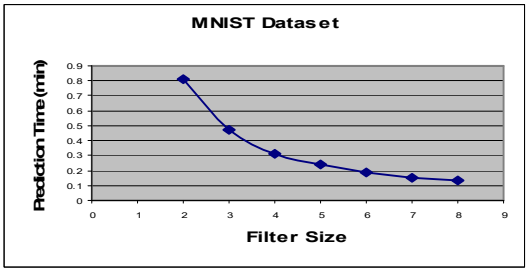


Figure 7v: Prediction Time Vs. Filter Size

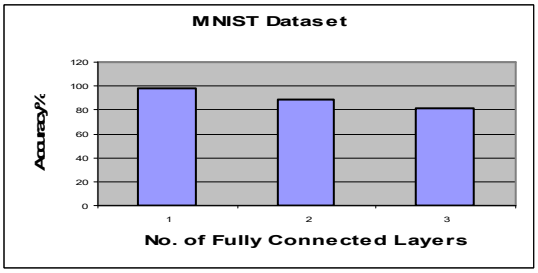


Figure 7w: Accuracy% Vs. # Conv. Layers

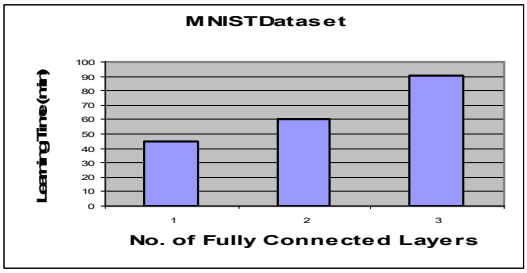


Figure 7x: Learning Time Vs. # Conv. Layers

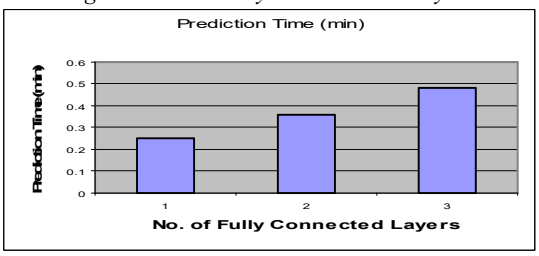


Figure 7y: Prediction Time Vs. # Conv. Layers

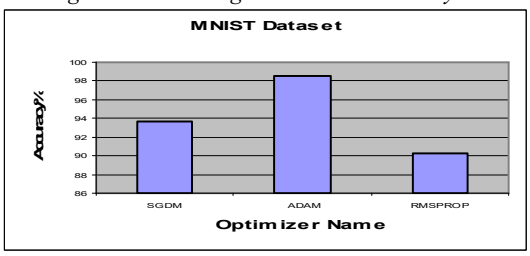


Figure 7z: Accuracy% for Optimizers

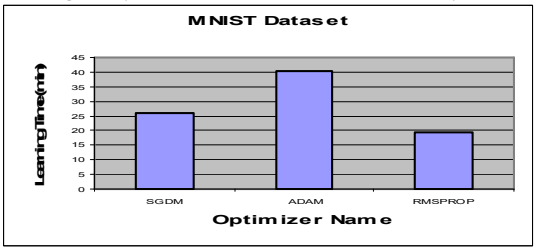


Figure 7aa: Learning Time for Optimizers

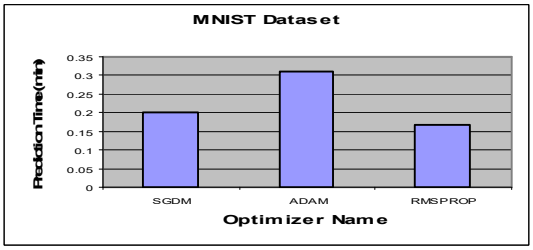


Figure 7ab: Prediction Time for Optimizers

number of epochs is an important parameter as it refers to the number of times the dataset of input training data is passed during training. Using too small or too large epochs may cause the network to present poor performance due to under-fitting orover-fitting as mentioned before.

Figure 6e presents the effect of using different batch sizes on the classification accuracy. The best accuracy value was 96.11%. Also, the learning time and prediction time were changed by changing the batch size as shown respectively in Figures 6g and 6h. Using a large batch size can change the dynamics of the training process and this increased the tuning time and cost. The batch size has an impact on the amount of parameter updates during training and also affects the learning time.

Figure 6i illustrates the effect of using multiple convolution layers. By increasing that number of layers, the accuracy was improved and the best accuracy occurred when using three convolution layers. The accuracy values were degraded when using extra number of layers. The number of convolution layers was changed from 1 to 5 layers.

The learning time and prediction time were increased by increasing the used layers as shown respectively in Figures 6k and 6l. It is easy to say that the number of convolution layers is effective as it can extract features. By increasing the number of convolution layers up to a certain value, the extracted features become increasingly concrete.

Figure 6m illustrates the effect of using different values of learning rates. The accuracy values were changed by changing the learning rate. The learning rate is important as it determines how much the model weights are adjusted with respect to the loss gradient. A small learning rate requires more updates to reach the minimum point of loss. Too large learning rates; on the other hand; may cause divergence from the optimal error point. Five different values of learning rates were operated and tested. The best accuracy value occurred when the adopted rate was 0.001 as shown in Figure 6m. Also, the learning time and prediction time were changed by changing the learning rate as shown respectively in Figures 6o and 6p.

Figure 6q shows the relationship between the number of used filters and accuracy values. Different number of filters was used and operated. The best accuracy was obtained when using 24

filters. The number of used filters ranges from 2 to 32. The accuracy was increased by increasing the number of filters till 24 filters then the accuracy was decreased. The learning time and prediction time were also affected by changing the number of filters as shown respectively in Figures 6r and 6s.

Figure 6t illustrates the effect of the filter size on the classification accuracy. Seven different values of filter sizes were operated. By increasing the filter size, the accuracy values were improved till a certain filter size then the performance was degraded as shown in figure 6x. The best accuracy occurred when using a filter of size 5x5. The adopted filter sizes were 2x2, 3x3, 4x4,… 8x8. The learning time and prediction time; on the other hand; were also affected by changing the filter size as shown respectively in Figures 6u and 6v.

Figure 6w illustrates the change of the accuracy values due to the change of the number of fully connected layers. Three fully connected layers were used and tested. The values of accuracy, learning time, and prediction time were changed as shown respectively in Figures 6w, 6x, and 6y by changing the number of fully connected layers.

Figures 6z, 6aa, and 6ab show respectively the change of accuracy values, learning time, and prediction time due to changing the optimizer type. Three optimizers were adopted mainly: SGDM, ADAM, and RMSPROP. Using ADAM optimizer, the best performance was presented as shown in Figure 6z. The learning time and prediction time were the highest values for ADAM optimizer.

Similarly, the same experiments were also done using the same machine and deep learning approaches on the MNIST dataset which was mentioned before. The results of the experiments are illustrated respectively in Figure 7a to 7ab. All the adopted hyper-parameters presented approximately the same behavior like what happened with the CIFAR-10 dataset but with different values. All the performance metrics were better for the MNIST dataset than those occurred for the CIFAR-10 dataset. This means that the dataset size, nature, and characterization play a vital role in the performance of the deep learning models. The values of learning time and prediction time for MNIST dataset were smaller and better than those values of CIFAR-10.

Moreover, the loss function was used to calculate the differences between the actual values and predicted ones. The loss values were also affected by the hyper-parameters mentioned above. This is shown respectively in Figures 6b, 6f, 6j, and 6n (for CIFAR-10 Dataset) and Figures 7b, 7f, 7j, and 7n (for MNIST Dataset).

In addition to the above, SVM was also operated on the CIFAR-10 and MNIST. SVM was chosen as it is one of the most common and promising supervised machine learning approaches. The most significant features were selected based on two feature selection methods: PCA and LDA. The behavior of SVM classifier was affected by the number of chosen features. For too small or too large number of features the accuracy values were not O.K. The best accuracy occurred when choosing the most significant features for CIFAR-10 and MNIST which were 70 and 125 respectively as shown in Figures 8a, and 9a. The learning time and prediction time were increased by increasing the number of features for the two adopted datasets but on different values as shown respectively in Figures 8b and 8c for CIFAR-10 dataset and 9b and 9c for MNIST dataset.

6. CONCLUSION

This work discussed the process of automatic image classification using machine and deep learning approaches. The convolution neural network (CNN); was adopted, analyzed and applied. The hyper-parameters of CNN were exploited to configure several CNN models. Such models were applied and tested using CIFAR-10 and MNIST datasets. Several CNN models were trained using different values of hyper-parameters such as number of convolutions, filter size, number of filters, number of epochs, batch size, learning rate, and others. The support vector machine (SVM) was also applied to classify images of the same datasets. The accuracy, learning time, and prediction time were used to evaluate the performance of the adopted deep and machine learning approaches.

From the experimental results, it is concluded that the CNN deep learning approach is computationally expensive and more complicated than SVM to enhance the image classification process. CNN is more powerful and dominant in image classification than that of the SVM. CNN is a better choice to achieve reliable accuracy of image classification compared with the SVM classifier. CNN can extract the feature maps from the 2D-images by using filters. High accuracy values between 92.22% and 98.77% were achieved with different learning time for different values of CNN hyper-parameters. The values of classification accuracy for CIFAR-10 were less than those corresponding values of MNIST using the same CNN models. The reason may be for the difficulty and complication of the CIFAR-10 images. The

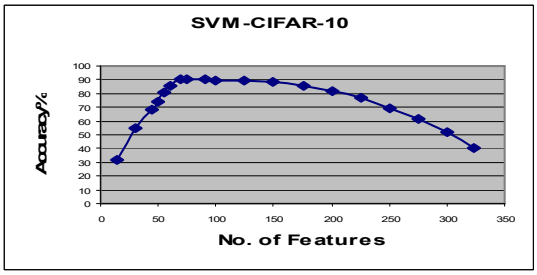


Figure 8a: Accuracy% Vs. No. of Features

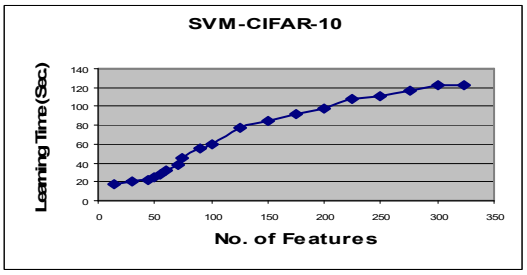


Figure 8b: Learning Time Vs. No. of Features

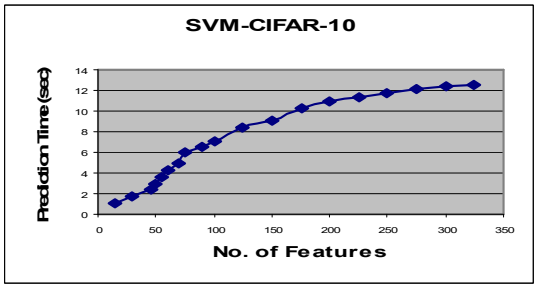


Figure 8c: Prediction Time Vs. No. of Features

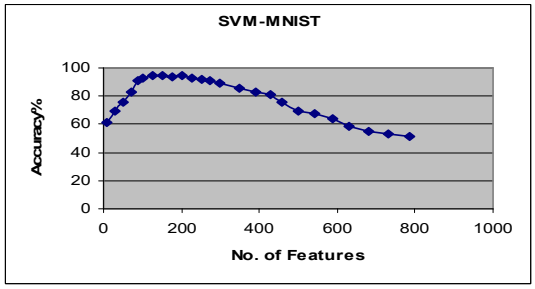


Figure 9a: Accuracy% Vs. No. of Features

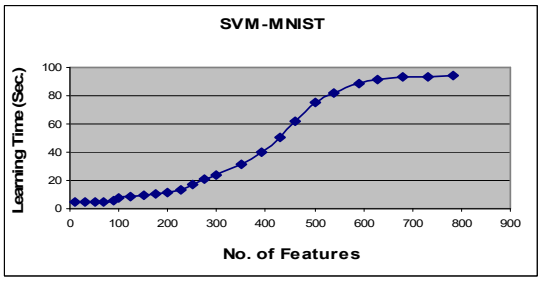


Figure 9b: Learning Time Vs. No. of Features

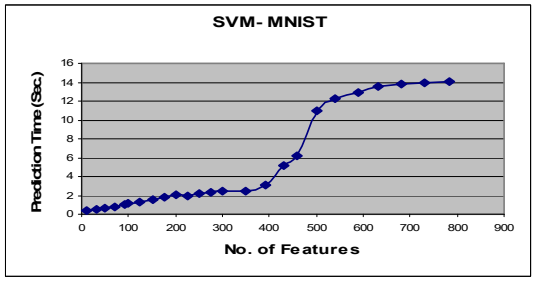


Figure 9c: Prediction Time Vs. No. of Features

occupied memory size and learning time were greater for CIFAR-10 dataset than the corresponding values of MNIST. Moreover, there is a trade-off between the learning time and accuracy values for image classification for both CIFAR-10 and MNIST. For the CIFAR-10, the accuracy values were in the range of 92.22% to 96.11%. For the MNIST dataset, the accuracy values were in the range of 96.33% to 98.77%. The CNN models achieved better accuracy values than those of the SVM by about 6% to 12%. The SVM; on the other hand; consumed smaller memory size and learning time compared with those of the CNN for classifying the images of the two adopted datasets. The running time for applying the CNN deep learning models was sometimes in the order of hours while their corresponding ones for the SVM were in the order of seconds or few of minutes sometimes.

All experiments were operated on a laptop supported by Windows-10, installed RAM 16.0 GB, and Processor Intel® Core ™ i5-3210M CPU@ 2.5GHZ processing speed. The experiments were implemented using Matlab-R2019a

Comparing the obtained results with some related efforts published in the literature by others; as shown in Table 2; refers that our results are better than the other ones. Different accuracy values for other research efforts were also registered using the same CNN deep learning models but on different datasets. The obtained results in this research work outperformed the other ones.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

REFERENCES:

[1] M. Manoj Krishna, M. Neelima, M. Harshali, and M. Venu Gopala Rao, "Image Classification using Deep Learning", The International Journal of Engineering& Technology, Vol. 7, No. 2, PP. 614-617, 2018.

[2] Vishali Aggarwal and Gagandeep, "A Review: Deep Learning Technique for Image Classification", The ACCENTS Transactions on Image Processing and Computer Vision, Vol. 4, No. 11, PP. 21-25, May 2018.

[3] Qinghe Zheng, Mingpiang Yang, Xinyu Tian, Nan Jiang, and Deqiang Wang, "A Full Stage Data Augmentation Method in Deep Convolutional Neural Network for Natural Image Classification", Hindawi, Discrete Dynamics in Nature and Society, Vol. 2020, Article ID 4706576, pp. 1-11, 2020, https://doi.org/10.1155/2020/4706576

[4] Hao Wu, Qi Liu, and Xiaodong Liu, "A Review on Deep Learning Approaches to Image Classification and Object Segmentation", Computers, Materials and Continia (CMC), Vol. 60, No. 2, PP. 575-597, 2019.

[5] Manu Gayal, Thomas Knackstedt, Shaofeng Yan, and Saeed Hassanpour, "Artificial Intelligence-Based Image Classification Methods for Diagnosis of Skin Cancer: Challenges and Opportunities", Computers in Biology and Medicine, Vol. 127, PP. 1-11, October 2020.

[6] B. Meena, K. Venkata, and S. Chittineni, "A Survey on Deep Learning Methods and Tools in Image Processing", The International Journal of Scientific and Technology Research, Vol. 9, Issue 2, PP. 1057-1062, Feb. 2020.

[7] Yoshihiro Shima, "Image Augmentation for Object Image Classification Based on Combination of Pre-Trained CNN and SVM", Journal of Physics, Conference Series 1004012001, PP. 1-8, 2018.

[8] Jun-e Liu and Feng-Ping An, "Image Classification Algorithm Based on Deep Learning-Kernel Function", Hindawi Scientific Programming, Vol. 2020, Article ID 7607612, PP. 1-14, 2020.

[9] Sehla Loussaief and Afef Abdelkrim, "Machine Learning Framework for Image Classification", The Advances in Science, Technology and Engineering Systems Journal, Vol. 3, No. 1, PP. 1-10, 2018.

[10] Priyanka Paygude, Rahul Garg, Pranjal Pthak, Abhinav Trivedi, and Aman Daj, "Image Processing Using Machine Learning", IJSDR, Vol. 5, Issue 9, PP. 471-477, September 2020.

[11] Md. Anwar Hossain and Md. Shahriar Alam Sajib, "Classification of Image Using Convolutional Neural Network (CNN)", The Global Journal of Computer Science and Technology: Neural and Artificial Intelligence, Vol. 19, Issue 2, Version 1.0, PP. 1-6, 2019.

[12] Muthukrishman Ramprasath, M. Vijay Anad, and Shanmugasundaram Hariharan, "Image Classification Using Convolutiuonal Neural Networkjs", The International Journal of Pure and Applied Mathematics, Vol. 119, No. 17, PP. 1307-1319, 2018.

[13] Fangming Chai and Kyoung-Don Kang, "Adaptive Deep Learning for Soft Real-Time.

[14] Image Classification", Technologies, License MDPI, Vol. 9, No. 20., PP. 1-23, 2021, https://www.mdpi.com/journal/technologies.

Image Datasets, Downloaded in 2021 From Internet From the Website https://www.mygrestlearning.com/blog/top-20-dataset-in-machine-learning/, 2021.

[15] Chen Wang and Yang Xi, "Convolutional Neural Network for Image Classification", The International Conference on Advanced Systems and Electric Technologies, Hammamet, Tunsiam, June 2018, Downloaded from the Website https://ieeexplore.ieee.org/document/8379889 .

[16] Samira Pouyanfar, Saad Sadiq, Yilin Yan, Haiman Tian, Yudong Tao, Maria Presa Reyes, Mei-Ling Shyu, Shu-Ching Chen and S.S. Iyengar, "A Survey on Deep Learning: Algorithms, Techniques, and Applications", The ACM Computing Surveys, Vol. 51, No. 5, Article 92, PP. -36, September 2018.

[17] Sunpreet Kaur and Sonika Jindal, "A Survey on Machine Learning Algorithms", The International Journal of Innovative Research in Advanced Engineering (IJIRAE), Vol. 3, Issue 11, PP. 6-14, November 2016.

[18] Shivam Singh, "Image Classification Using Deep Learning", Downloaded from the Website https://ijcrt.org/papers/IJCRT_195054.pdf, PP. 1-8, 2021.

[19] Songshang Zou, Wenshu Chen, and Hao Chen, "Image Classification Model Based on Deep Learning in Internet of Things", Hindawi-Wireless Communications and Mobile

Computing, Vol. 2020, Article-ID 6677907, PP. 1-16, 2020, https://doi.org/10.1155/2020/6677907.

[20] N. Banupriya, S. Saranya, Rashmi Swaminathan, Sanchithaa Harikumar, Sukitha Palanisamy, "Animal Detection Using Deep Learning Algorithm", Journal of Critical Reviews, Vol. 1, Issue 1, PP. 434-439, 2020.

[21] Subarna Shakya, "Analysis of Artificial Intelligence-Based Image Classification Techniques", Journal of Innovative Image Processing (JIIP), Vol. 2, No. 1, PP. 44-54, 2020.

[22] R. Thillaikkarasi and S. Sarovanan, "An Enhancement of Deep Learning Algorithm for Brain Tumor Segmentation Using Kernal-Based CNN with M-SVM", Journal of Medical Systems, Image and Signal Processing, PP. 43-84, 2019, https://doi.org/10.1007/s10916-019-1223-7.

[23] Hung Dao, "Image Classification Using Convolutional Neural Networks", Bachelor's Thesis Presented to Information Technology Dept., Oulu University of Applied Sciences, Spring 2020.

[24] Muhammad Uzair and Noreen Jamil, "Effects of Hidden Layers on the Efficiency of Neural Networks", The 23rd IEEE International Multi-topic Conference, PP. 1-6, 2020, DOI: 10.1109/INMIC50486.2020.9318195.

[25] Saahil Afaq and Smith Rao, "Significance of Epochs on Training a Neural Network", The International Journal of Scientific & Technology Research, Vol. 9, Issue 06, PP. 485-488, June 2020.

[26] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep Learning for Computer Vision: A Brief Review", Computational Intelligence and Neuroscience, PP. 1–13, 2018. DOI: 10.1155/2018/ 706834.

[27] A. Latif, A. Rasheed, U. Sajid, J. Ahmed, N. Ali, N. I. Ratyal, B. Zafar, S. H. Dar, M. Sajid, and T. Khalil, "Content-based Image Retrieval and Feature Extraction: A Comprehensive Review", Mathematical Problems in Engineering, PP. 1–21, August 2019. https://doi.org/10.1155/2019/ 965835.

[28] S. Chaudhry, and R. Chandra, (2016, October). "Unconstrained Face Detection from a Mobile Source Using Convolutional Neural Networks", Lecture Notes in Computer Science, 9948, PP. 567–576, October 2016. Sub-series Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics. https://link.springer.com/chapter/10.1007%2F978-3-319-46672-9_63.

[29] I. Gogul, and V.S. Kumar, "Flower Species Recognition System Using Convolution Neural Networks and Transfer Learning", The Fourth International Conference on Signal Processing, Communication and Networking (ICSCN), March, PP. 1–6, March 2017. Chennai, India.

[30] Mehmood ulHasan, Saleem Ullah, Muhammad Jaleed Khan, Khurram Khurshid, "Comparative Analysis of SVM, ANN, and CNN for Classifying Vegetation Species Using Hyperspectral Thermal Infrared Data", The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XLII-2/W13, PP. 1861-1868, June 2019.

[31] Yoshihiro Shima, "Image Augmentation for Object Image Classification Based on Combination of Pre-Trained CNN and SVM", Journal of Physics: Conference Series 1004 (2018) 012001, PP. 1-8, 2018.

[32] Pooja Kamavisdar, Sonam Saluja, Sonu Agrawal, "A Survey on Image Classification on Approaches and Techniques", The International Journal of Advanced Research in Computer and Communication Engineering, Vol. 2, Issue 1, PP. 1005-1009, January 2013.

[33] Shanqing Gu, Manisha Pednekar, and Robert Slater, "Improve Image Classification Using Data Augmentation and Neural Networks", SMU Data Science Review, Vol. 2, No. 2, PP. 1-43, 2019, Available at https://scholar.smu.edu/datasciencereview/vol.2/iss2/iss2/1.

[34] Linjian Ma, Gabe Montague, and Jiayu Ye, "Inefficiency of K-FAC for Large Batch Size Training", Downloaded From the Internet in 2022 From arXiv:1903.06237v3 [cs.LG] 31 Jul 2019.

[35] Abe Winters, "Examining the Effect of Hyper Parameters on the Training of a Residual Network for Emotion Recognition", The 32nd Twente Student Conference on IT, Enschede, The Netherland, Jan 31, 2020.

[36] Jinia Konar, Prerit Khandelwal, and Rishabh Tripathi, "Comparison of Various Learning Rate Scheduling Techniques on Convolution Neural Network", The IEEE International Students Conference on Electrical, Electronics and Computer Science (SCEECS-2020), PP. 1-5, 2020. Downloaded From the Internet in 2022.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[37] D. Lavanya and D. Usha Rani, "Analysis of Feature Selection with Classification: Breast Cancer Datasets", The Indian Journal of Computer Science and Engineering (IJCSE), Vol. 2, No. 5, PP. 756-763, Oct-Nov 2011.

[38] Mahdieh Labani, Parhan Moradi, Fardin Ahmadizar, and Mahdi Jalili, "A Novel Multivariate Filter Method for Feature Selection in Text Classification Problems", Journal of Engineering Applications of Artificial Intelligence, Vol. 70, PP. 25-37, 2018.

[39] Zeshan Hussain, Francisco Gimenez, Darvin Yi, and Daniel Rubin, "Differential Data Augumentation Techniques for Medical Imaging Classification Tasks", Downloaded From the Website https;//www.researchgate.net/325532618, PP. 979-984, May 2020.

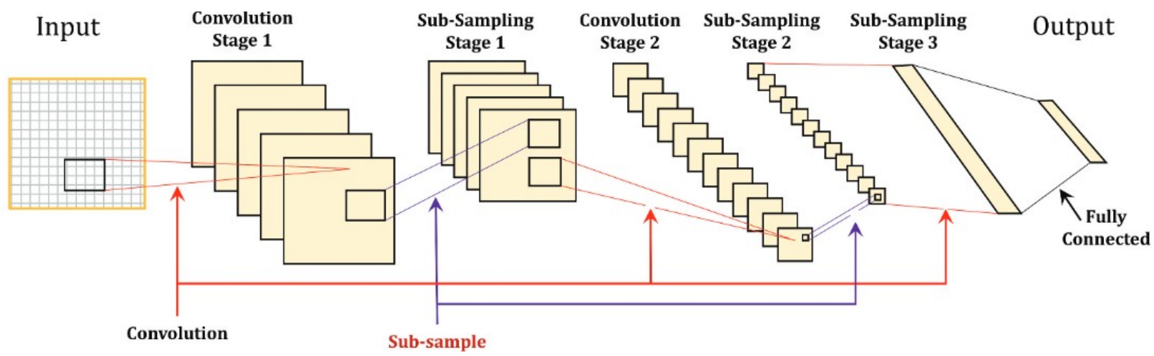


Figure 1: Main Outlines of the CNN Architecture [26], [27], [28], [29]

Table 1: Setting Values Of The Hyper Parameters For Image Classification

Parameter Name	Setting Values
Number of Filters Used	{2, 4, 8, 16, 24, 32}
Filter Size	{2x2, 3x3, 4x4, 5x5, 6x6, 7x7, 8x8}
Filter Shape	{Padding, Stride=1, Stride=2, and Stride=3}
Number of Convolution Layers	{1, 2, 3, 4, 5}
Number of Fully Connected Layers	{1, 2, 3}
Learning Rate	{0.1, 0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001}
Optimizers' Names	{SGDM, ADAM, RMSPROP}
Batch Size	{10,000; 15,000; 20,000; 25, ………… 50,000}
Number of Epochs	{10, 20, 30, 40, ……….. 130}
Number of Neurons	{16, 32, 48, 64, 80, 96, 112, 128, 144, 160, 176}

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	11	150	253	202	31	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	37	251	251	253	107	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	21	197	251	251	253	107	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	110	190	251	251	251	253	169	109	62	0	0	0	0
0	0	0	0	0	0	0	0	0	0	253	251	251	251	251	253	251	251	220	51	0	0	0
0	0	0	0	0	0	0	0	0	182	255	253	253	253	253	234	222	253	253	253	0	0	0
0	0	0	0	0	0	0	0	63	221	253	251	251	251	147	77	62	128	251	251	105	0	0
0	0	0	0	0	0	0	32	231	251	253	251	220	137	10	0	0	31	230	251	243	113	5
0	0	0	0	0	0	0	37	251	251	253	188	20	0	0	0	0	0	109	251	253	251	35
0	0	0	0	0	0	0	37	251	251	201	30	0	0	0	0	0	0	31	200	253	251	35
0	0	0	0	0	0	0	37	253	253	0	0	0	0	0	0	0	0	32	202	255	253	164
0	0	0	0	0	0	0	140	251	251	0	0	0	0	0	0	0	0	109	251	253	251	35
0	0	0	0	0	0	0	217	251	251	0	0	0	0	0	0	21	63	231	251	253	230	30
0	0	0	0	0	0	0	217	251	251	0	0	0	0	0	0	144	251	251	251	221	61	0
0	0	0	0	0	0	0	217	251	251	0	0	0	0	0	182	221	251	251	251	180	0	0
0	0	0	0	0	0	0	218	253	253	73	73	228	253	253	255	253	253	253	253	0	0	0
0	0	0	0	0	0	0	113	251	251	253	251	251	251	251	253	251	251	251	147	0	0	0
0	0	0	0	0	0	0	31	230	251	253	251	251	251	251	253	230	189	35	10	0	0	0
0	0	0	0	0	0	0	0	62	142	253	251	251	251	251	253	107	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	72	174	251	173	71	72	30	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 5 (a): Representation of the "O" Image (An Example from MNIST Dataset)

Journal of Theoretical and Applied Information Technology
15th December 2022. Vol.100. No 23

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

B Representation of the "Bird" image

130	128	131	138	144	151	159	163	166	166	160	151	143	138	145	154	158	157	155	150	143	140	138	136	131	124	122	128	131
119	120	124	128	135	140	146	152	160	164	157	149	145	146	152	157	164	165	160	153	146	140	136	135	130	123	117	117	122
120	121	122	131	141	156	173	189	180	158	149	147	148	154	159	166	171	170	161	153	150	145	140	136	128	123	116	113	122
118	125	139	172	199	214	226	234	215	180	176	181	176	172	169	172	172	163	155	151	149	145	142	139	132	129	127	125	129
137	175	210	235	239	236	230	222	222	224	229	228	222	207	188	174	165	148	142	144	143	140	136	133	133	134	136	136	138
207	232	237	231	224	218	215	210	218	226	224	215	204	187	190	184	154	138	129	128	133	137	131	127	133	136	134	140	143
233	229	224	214	210	203	197	193	203	210	210	201	192	182	166	168	159	136	127	130	134	125	114	122	135	139	140	147	147
219	215	210	201	196	190	180	181	191	203	201	189	180	175	167	142	145	134	111	117	110	81	78	83	97	145	144	149	149
207	204	197	187	184	180	173	173	171	171	155	135	113	109	134	117	106	98	76	76	68	69	81	59	74	152	152	150	150
197	192	184	165	164	167	164	158	152	112	78	90	75	67	68	77	94	121	106	80	62	58	63	68	103	134	183	163	150
180	177	167	156	153	150	152	145	115	96	99	83	79	64	59	75	103	115	99	63	60	65	71	95	76	91	203	186	161
162	161	154	161	155	147	142	115	74	104	160	122	87	69	47	78	107	87	92	74	71	63	74	60	62	93	177	198	176
142	141	147	158	148	133	115	81	62	82	88	113	75	57	48	74	88	90	109	88	63	63	73	52	65	60	113	196	186
125	127	142	136	95	87	122	96	80	92	74	59	52	70	57	67	84	95	115	94	54	41	59	72	44	34	48	123	178
117	122	119	89	69	88	96	77	80	70	88	65	67	104	75	62	78	94	107	92	53	51	68	70	52	45	37	74	140
117	110	85	75	85	86	73	75	69	45	55	60	95	102	105	78	66	77	104	104	63	64	88	71	48	72	58	55	114
114	89	67	74	100	82	79	86	66	46	78	83	82	109	97	76	84	82	113	99	80	87	78	85	55	53	79	66	96
100	77	62	58	71	86	101	95	61	62	114	93	65	98	86	79	127	94	112	103	79	99	72	92	62	39	74	76	86
81	60	64	85	60	99	104	89	62	104	101	63	84	117	82	94	134	76	93	101	60	75	54	86	97	110	113	80	78
82	61	88	102	81	108	107	76	78	137	85	52	81	130	98	72	83	66	79	85	79	94	109	117	128	132	121	78	59
90	74	96	107	98	84	92	77	89	123	99	113	140	150	135	126	113	99	98	104	101	104	112	107	93	82	77	68	70
88	79	98	106	90	62	60	91	89	140	165	169	162	141	131	126	105	98	91	90	98	106	116	115	88	80	77	65	58
78	67	61	65	61	66	71	86	116	133	102	105	117	122	129	126	109	99	97	102	114	118	114	112	100	98	85	65	48
92	58	37	67	69	68	96	97	103	103	105	123	139	139	141	135	118	107	102	104	105	107	106	105	117	103	84	69	59
96	72	42	48	47	62	113	123	125	154	153	138	138	138	128	118	109	105	100	96	91	92	94	111	132	106	80	69	62
111	118	106	67	55	76	91	116	134	162	173	152	135	126	113	100	96	92	84	76	76	88	102	115	124	102	83	84	74
166	179	156	107	69	74	81	82	114	156	164	141	126	121	112	92	82	67	69	78	99	113	105	102	102	89	66	78	78
192	185	164	123	70	61	66	49	52	92	123	130	124	119	113	102	64	55	79	92	98	88	78	72	68	79	102	125	141
193	184	171	138	88	45	57	67	40	53	56	74	87	84	82	81	50	39	60	65	60	58	56	41	55	105	135	158	178
193	181	174	159	117	59	47	61	77	94	58	38	37	43	49	50	47	48	52	54	57	51	41	30	72	115	144	166	185
198	176	169	170	137	100	54	84	130	105	79	64	55	33	31	38	53	62	62	60	50	39	32	40	88	119	146	164	180
197	178	156	163	151	128	77	79	114	94	92	62	73	72	41	32	44	58	55	47	39	38	33	52	97	119	141	157	170

R Representation of the "Bird" image

162	164	169	179	185	192	200	204	209	210	204	193	184	180	187	198	202	202	199	194	188	185	184	184	179	172	170	176	179	184	190	202
153	158	164	169	176	181	187	195	203	208	200	190	186	186	192	198	205	206	202	197	190	184	182	182	177	170	164	163	169	178	189	201
157	161	164	167	175	191	205	214	213	201	195	189	190	196	200	204	210	207	201	197	194	189	186	184	175	170	163	160	169	178	190	202
154	159	171	195	221	238	248	251	239	214	211	209	206	203	201	207	210	205	199	196	194	191	188	185	180	179	174	171	175	183	194	201
167	197	227	248	255	255	254	253	250	247	248	245	242	230	214	206	204	196	192	192	192	190	187	184	185	187	183	180	183	187	191	193
232	250	254	254	251	249	250	251	253	254	253	253	247	233	232	219	192	180	177	180	184	187	184	182	184	184	179	184	187	186	186	184
255	254	251	252	252	250	246	244	249	249	250	253	248	241	222	215	204	177	173	179	179	165	153	164	179	183	184	191	191	186	185	187
254	251	248	249	249	246	237	238	246	251	246	240	234	233	225	199	196	177	153	156	143	107	97	105	129	186	188	193	193	188	189	191
251	245	240	237	236	233	226	231	238	229	193	162	143	143	169	149	135	125	97	90	80	79	89	72	99	187	193	193	194	191	193	194
247	238	231	223	224	227	224	222	215	156	96	97	85	81	83	92	109	137	117	84	67	63	69	80	123	162	219	202	192	191	198	200
237	230	221	221	220	217	219	207	156	112	101	86	82	66	65	87	115	127	108	68	65	70	78	107	94	114	231	219	200	188	192	193
226	221	214	221	215	208	201	155	88	105	167	133	96	75	51	84	114	94	98	79	77	68	83	73	78	110	197	225	212	195	181	183
216	209	206	205	190	171	148	98	63	83	101	132	92	71	56	79	92	95	113	92	68	68	82	64	76	70	128	223	223	203	182	177
211	199	190	172	120	98	124	102	86	99	83	74	67	85	67	71	88	99	117	96	58	47	68	82	50	37	56	151	220	220	199	181
200	185	158	115	87	96	99	81	85	74	94	77	80	116	84	66	82	98	110	94	57	57	77	80	57	47	40	92	179	214	217	191
188	155	108	87	95	94	79	77	70	46	59	70	105	112	113	82	70	82	107	106	67	69	97	81	54	75	62	65	140	204	220	207
170	114	72	77	104	88	85	86	66	45	80	91	90	118	104	80	88	86	116	101	85	93	86	95	61	56	92	75	105	166	215	218
128	87	65	60	75	92	106	97	61	62	114	93	67	102	90	81	129	97	116	110	87	108	81	100	69	45	92	90	93	118	180	223
96	68	68	90	65	104	109	92	65	108	102	60	84	120	86	99	139	80	101	114	72	88	69	101	113	126	134	97	88	95	143	209
96	71	91	106	86	113	112	81	84	143	90	52	84	136	106	82	93	76	92	102	96	111	131	142	153	157	145	94	69	72	121	176
102	85	98	111	104	89	98	86	98	133	106	116	145	158	146	140	126	112	115	126	123	125	136	133	119	108	102	84	79	109	129	167
101	93	100	106	93	65	64	100	99	152	177	178	172	153	145	142	122	115	110	113	120	129	141	141	112	103	98	79	67	98	119	146
98	85	68	62	57	64	69	87	123	147	121	123	135	140	146	141	127	120	117	121	133	137	141	145	125	118	100	76	55	77	100	91
117	80	51	70	69	68	95	99	110	116	121	140	156	156	158	153	138	129	125	127	128	130	136	138	143	123	97	76	62	56	91	93
126	98	65	59	53	65	115	127	131	161	163	153	153	153	144	137	130	128	127	124	119	120	126	143	158	126	96	81	71	45	59	80
145	148	137	86	67	82	94	121	138	165	178	165	149	139	129	122	119	118	113	109	109	120	136	147	150	122	106	107	94	78	60	82
195	209	190	140	89	83	87	87	117	157	170	158	141	133	125	111	101	86	92	105	125	138	137	133	126	125	109	121	119	144	160	161
215	215	201	171	105	73	67	52	56	96	131	144	140	134	129	119	81	72	100	118	124	113	111	104	102	134	158	177	191	210	226	224
215	215	210	191	138	68	54	71	48	64	67	82	101	103	103	101	70	58	84	94	89	87	86	77	107	172	197	211	226	235	237	238
214	213	213	201	173	102	58	68	89	112	77	51	51	58	68	72	69	69	76	79	83	77	61	63	134	188	209	220	232	240	241	242
218	209	208	200	191	157	87	97	139	125	103	86	74	48	46	59	74	84	85	82	71	58	43	73	156	192	213	220	228	232	236	239
216	207	201	199	192	178	132	107	107	109	112	90	103	100	61	45	59	79	80	70	57	50	43	96	168	193	212	220	226	231	233	235

Cont. G Representation of the "Bird" image

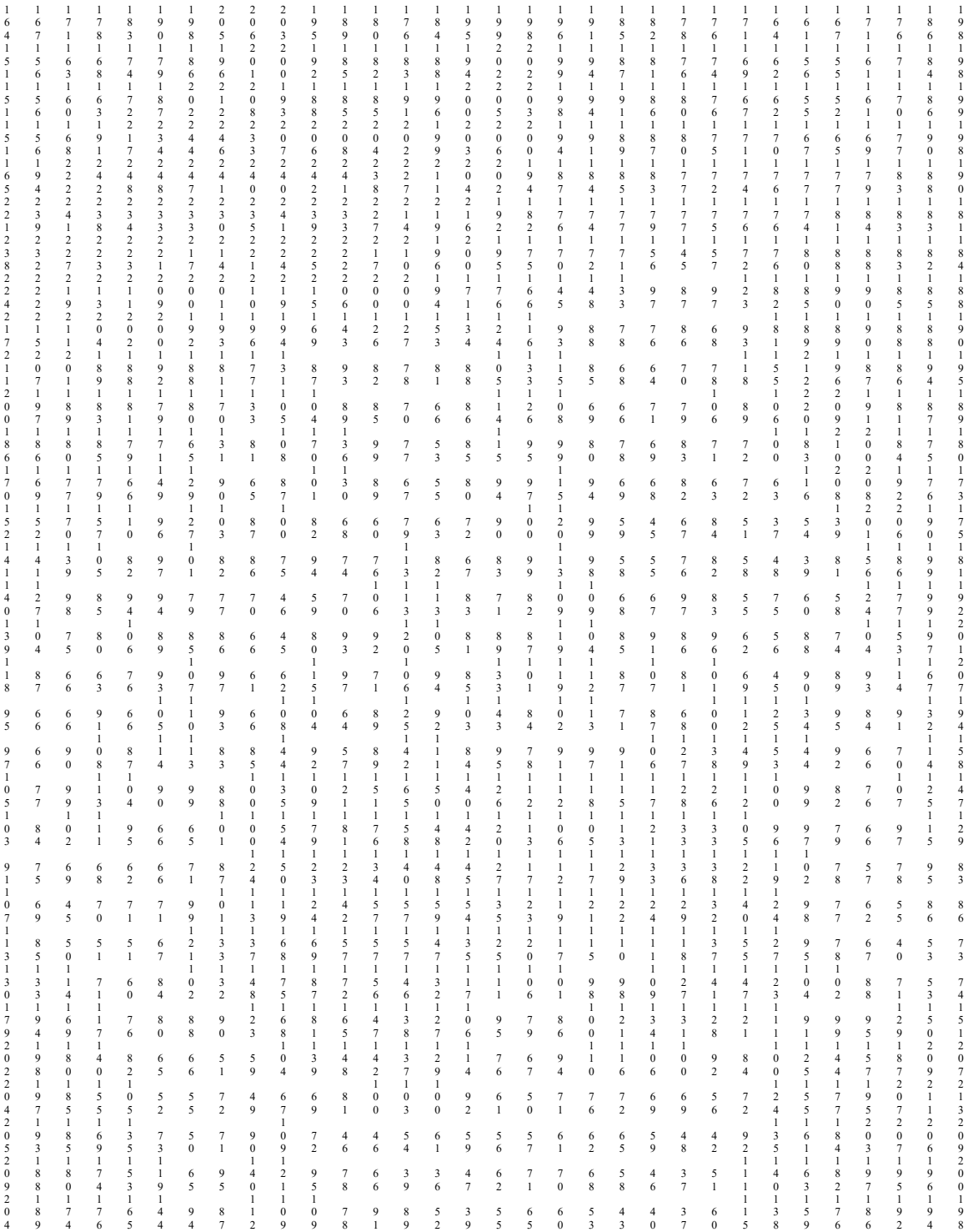


Figure 5 (b): Representation of the "Bird" Image (An Example from CIFAR-10 Dataset)

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 2: The Obtained Results VS. Some Other Related Ones Published in the Literatures

Applying CNN Models on CIFAR-10 and MNIST Datasets	**Accuracy%**	
Authors	**CIFAR-10 Dataset**	**MNIST Dataset**
[36]	76.28% to 88.22%	95.16% to 97.89%
[13]	80% to 89%	95% to 99%
[11]	76.82% to 93.47%	----------
[25]	----------	90% to 99.38%
[This work, 2022]	92.22% to 96.11%	96.33% to 98.77%
Applying CNN Models on Different Datasets		Accuracy%
[20]	89.17% Food-101	94% Places-2 Dataset
[35]	91.65% to 96.75% RAF-DB	Real-World Affective Faces Database
[39, et. al., ……..]	84% to 88% DDSM	Digital Database for Screening Mammography