ISSN (online):2396-9889

INTERNATIONAL RESEARCH JOURNAL OF TECHNOLOGY AND APPLIED SCIENCE

http://www.irjtas.com

(An ISO Certified Journal)

VOL. 05 Issue 02 FEBRUARY 2021

An Enhanced Convolutional Neural Network on Image Classification over CIFAR-10 Dataset

Megha Sharma
Mtech Scholar
siasharma815@gmail.com

Prof. Amit Ganguli
M. Tech. Co-ordinator
amitganguli@sistec.ac.in

Prof. Ajit Kumar Shrivastava
Head of Dept, CSE
SISTecR.hodcs@sistec.ac.in

***Abstract*— Visual beholding is of great importance for humans to interact with each other and thus the wildlife . We possess a huge ability of visual recognition as we'll almost effortlessly recognize objects encountered in our life like animals, faces, and food. Especially, humans can easily recognize an object albeit it's getting to vary in position, scale, pose, and illumination. Such ability is known as core beholding , and is run through the ventral stream within the human sensory system. Yet, since past few years, machine performance has been dramatically improved due to the reemergence of convolutional neural networks (CNN) and deep learning, and thus even surpasses human performance. Like customary neural organizations, which are enlivened by natural neural frameworks, the design of CNNs for viewing is feed forward and comprises of a few layers during a various leveled way. Particularly, some works reveal hierarchical correspondence between CNN layers and other people within the human beholding system. beholding performance of deep neural networks is typically measured on datasets commonly utilized within the sector like CIFAR100, and CIFAR10. In this dissertation we have taken convolution neural network with different layers configuration and apply on CIFAR-10 dataset and we found that our modified CNN model takes more time but perform better as compared to Traditional CNN.**

Keywords*— *artificial neural networks; cifar-10; classification; image; convolutional neural networks; keras; python;jupyter notebook;machine learning.

I.INTRODUCTION

The ability to classify things correctly requires many hours of training. People get things wrong many times, until eventually, they get it right. The same structure applies to machine learning. By using a high-quality set of data, deep learning can classify objects comparatively well or even better than humans can. With achieving utterly accurate image classifier, some of the monotonous jobs could be replaced by machines, so that humanity could focus on the most enjoyable activities.

Achieving high classification rate on a set of tiny images tends to be difficult, as some of the features that identify specific class are barely visible even to human eyes. The area of computing vision is under constant development in order to be the most effective in investigating and successfully classifying every kind of object. This type of analysis could advance, for example, the usefulness of autonomous cars, which tend to be ineffective in particular situations of object recognition, leading to significant damages. Most of the traditional neural network algorithms do not achieve as satisfying results to be acceptable for most available jobs. The indicated fact disqualifies machines from replacing the monotonous human activities.

This project implements the structure of CNNs different from traditional, where it performs classification on 10 classes of multiple, evenly distributed images available in the CIFAR- 10 dataset. The improved model replaces the max-pooling and dense function with two-dimensional convolution layers, with the achievement of higher classification rate, basing its structure on the model .

ISSN (online):2396-9889
INTERNATIONAL RESEARCH JOURNAL OF TECHNOLOGY AND APPLIED SCIENCE
http://www.irjtas.com
(An ISO Certified Journal)
VOL. 05 Issue 02 FEBRUARY 2021

II. LITERATURE REVIEW

According to [1], — The conventional convolutional neural organization ordinarily introduces the loads of all organization layers all at once before network preparing, and afterward refreshes the loads of the organization by back-spread calculation to improve the exactness of the organization during network preparing. Nonetheless, with the expansion of organization profundity, the computational expense of this strategy will increment significantly and the test precision will be influenced. To tackle this issue, a strategy for steadily reinitializing the loads of each layer is proposed, that is, after a specific preparing period, the heaviness of the past layer is resolved and stay unaltered, at that point instate the loads of every ensuing layer, rehash this progression until the loads of all layers are resolved. To check the presentation of the technique, a progression of tests were done on the CIFAR10 dataset. The outcomes show that the exactness of the organization is improved by 9% and the preparation time is decreased by 29%. It shows that the technique can improve the precision of the organize and decrease the preparation time.

In [2], Training the deep learning models involves learning of the parameters to meet the objective function. Typically normally the goal is to limit the misfortune caused during the learning interaction. In a directed method of learning, a model is given the information tests and their individual results. At the point when a model creates a yield, it contrasts it and the ideal yield and afterward takes the distinction of produced and wanted yields and afterward endeavors to carry the created yield near the ideal yield. This is accomplished through improvement calculations. A streamlining calculation experiences a few cycles until assembly to improve the precision of the model. There are a few sorts of streamlining strategies created to address the difficulties related with the learning cycle. Six of these have been taken up to be analyzed in this investigation to acquire experiences about their complexities. The strategies researched are stochastic inclination plunge, nesterov energy, rmsprop, adam, adagrad, adadelta. Four datasets have been chosen to play out the analyses which are mnist, fashionmnist, cifar10 and cifar100. The ideal preparing results acquired for mnist is 1.00 with RMSProp and adam at age 200, fashionmnist is 1.00 with rmsprop and adam at age 400, cifar10 is 1.00 with rmsprop at age 200, cifar100 is 1.00 with adam at age 100. The most noteworthy testing results are accomplished with adam for mnist, fashionmnist, cifar10 and cifar100 are 0.9826, 0.9853, 0.9855, 0.9842 individually. The examination of results shows that adam improvement calculation performs better compared to others at testing stage and rmsprop and adam at preparing stage.

In [3], Training neural networks is a computationally challenging problem that requires significant time efforts. In this paper, we propose two approaches that improve efficiency of this task by actively selecting most relevant points from a training data set. The first approach forms a batch that maximizes the reduction of the estimator's entropy, while the second approach only trains on data points whose predicted probability is below a predetermined threshold. Both techniques rely on data metrics to speed up training while retaining the epoch based neural network training framework. The results demonstrate that the proposed methods enable significant reduction of training time in experiments on the CIFAR10 dataset without compromising the accuracy.

III PROBLEM DEFINITION

CIFAR-10 is publicly available online.[3] The dataset consists of 60,000 32x32 color images used for visual perception . We keep the split of train and test set within the official data. There are 50,000 images within the training set and 10,000 within the test set. additionally , there are 10 object classes in total and one image belongs to a particular class. There are not any intersections among the ten classes. The label classes are namely airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck. Both the preparation and test set are named for preparing and testing. Notably, the dataset from Kaggle is different from the web offical dataset in [3] therein the organizers mixed some junk data within the test set so as to ensure the justice of the competition. Since we'll not submit the result on Kaggle, we use the

ISSN (online):2396-9889
INTERNATIONAL RESEARCH JOURNAL OF TECHNOLOGY AND APPLIED SCIENCE
http://www.irjtas.com
(An ISO Certified Journal)
VOL. 05 Issue 02 FEBRUARY 2021

first test set without junk data on to provide an estimation of our prediction accuracy.

CIFAR-10 gives us natural color images. the colour information might not be very essential in some algorithms. However, if we apply the only thanks to process the info (averaging RGB to grayscale), we'll definitely lose information. Moreover, in natural images, color could also be associated with different objects. for instance , flowers tend to possess bright warm colors while trucks have relatively cold colors. Hence, we determined to not convert the pixels to grayscale.

IV PROPOSED WORK

The CIFAR-10 dataset contains of 60,000 32x32 color images of 10 classes, with 6000 foreach class . Over the years, tons of works are reported regarding the image classification problem with CIFAR-10 dataset . the very best accuracy thus far is achieved by using modified convolutional neural network(CNN) with fractional max-pooling . during this dissertation, the CIFAR-10 dataset was divided into 50,000 labeled training images and 10,000 unlabeled testing images. We further divided the training set into 49,000 training samples and 1,000 validation samples to pick the simplest model and hyper parameters. The classifier trained on the 49,000 samples with the optimized parameters was then wont to predict on the testing set and evaluate the prediction accuracy.

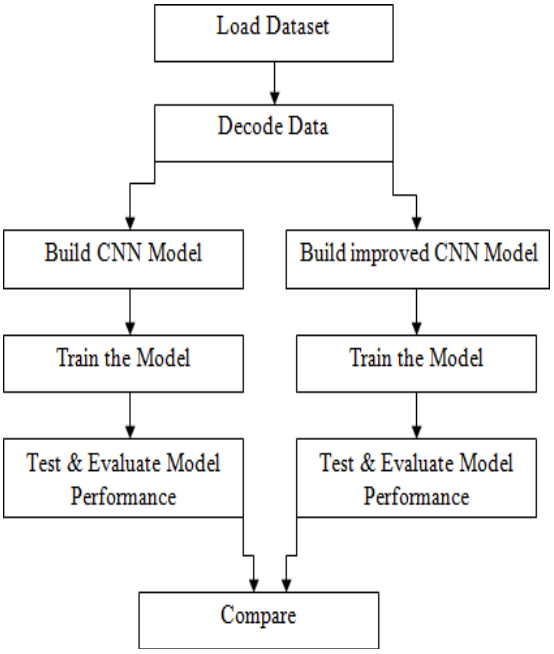

Figure 1. Proposed Block Diagram

Step 1: First we collect the CIFAR-10 image dataset from web.

Step 2: The dataset we have got is encoded so before performing further operations we first performing the decoding operation on the dataset.

Step 3: Now we trained two CNN model, Traditional and Modified CNN model.

Step 4: We can apply test dataset on both the models and evaluate the performance of the models.

V EXPERIMENTAL & RESULT ANALYSIS

Throughout this dissertation we employ two subsets of the 80 million tiny images dataset [2]. The 80 million tiny images dataset may be a collection of 32 × 32 color images obtained by searching various online image search engines. For experimental analysis, we use the CIFAR-10 dataset, which may be a labeled subset of the 80 million tiny images, containing 60,000 images. It contains ten classes: plane, auto, bird, feline, deer, canine, frog, pony, boat, and truck. Each class contains precisely 6,000 pictures during which the predominant item inside the picture is of that class. We are using Jupyter notebook for performing experimental analysis on the CIFAR-10 image classification.

ISSN (online):2396-9889
INTERNATIONAL RESEARCH JOURNAL OF TECHNOLOGY AND APPLIED SCIENCE
http://www.irjtas.com
(An ISO Certified Journal)
VOL. 05 Issue 02 FEBRUARY 2021

Data Preprocessing

First we Import all the necessary packages into jupyter notebook showing figure 2. The main function for CNN used is keras which is an advance library for building unsupervised machine learning model.

```
%matplotlib inline
import matplotlib
from matplotlib import pyplot as plt
import numpy as np
from keras.callbacks import ModelCheckpoint
from keras.layers import Lambda, Conv2D, MaxPool
from keras.models import load_model, Sequential
from keras.optimizers import Adam
```

Figure 2. Import of packages

After that, we are import the ―decoder.py‖ file that contains a gaggle of functions written for decoding the CIFAR-10 dataset using the Pickle package and plotting the labels and pictures data into arrays using the NumPy library. The next step requires using the imported functions from the ―decoder.py‖ file to load class names, check the quantity of classes and define the size of the input image, which is 32 by 32 pixels, and specify the quantity of channels to be three (red, green and blue). The three arrays of numbers contains values from 0 to 255 what indicates the pixel intensity at that point . With this information, the CNN can describe the probability of an object being of a specific class.

Another significant operation is to decode and fetch the photographs . The labels of classes are using integer data type, whereas class is using one-hot encoded vectors. the entire CIFAR-10 data divides into two sets – 83% (50000) of images for training and remainder 17% (10000) for testing, shown in figure 3.

Load the training dataset. Labels are integers whereas class is one-hot encoded vectors.

```
images_train, labels_train, class_train = get_train_data()
```

```
Decoding file: data/data_batch_1
Decoding file: data/data_batch_2
Decoding file: data/data_batch_3
Decoding file: data/data_batch_4
Decoding file: data/data_batch_5
```

Normal labels - Integers

```
print(labels_train)
```

```
[6 9 9 ..., 9 1 1]
```

Classes - One hot encoded labels

```
print(class_train)
```

```
[[ 0.  0.  0. ...,  0.  0.  0.]
 [ 0.  0.  0. ...,  0.  0.  1.]
 [ 0.  0.  0. ...,  0.  0.  1.]
 ...,
 [ 0.  0.  0. ...,  0.  0.  1.]
 [ 0.  1.  0. ...,  0.  0.  0.]
 [ 0.  1.  0. ...,  0.  0.  0.]]
```

Load the testing dataset.

```
images_test, labels_test, class_test = get_test_data()
```

```
Decoding file: data/test_batch
```

Print the size of the training and testing set.

```
print("Training set size:\t",len(images_train))
print("Testing set size:\t",len(images_test))
```

```
Training set size:      50000
Testing set size:       10000
```

Figure 3. Process of decoding and fetching data

Keras library makes building models very intuitive since every layer are often defined in one line of code using ―model.add()‖ function. The code used for the whole operation is self-explanatory and summarises the CNN model using four two-dimensional layers after executing the function shown in Figure 4. Firstly, the model is initialised employing a sequential function, which allows building a linear stack of layers treated as a stack of objects, where each of them passes data to subsequent one. the primary two ―Conv2D(32, (3, 3)‖ scripts initialise 32 convolution filters of 3x3 size each, after which another two ―Conv2D(64, (3, 3)‖ scripts use increased number of filters.

ISSN (online):2396-9889
INTERNATIONAL RESEARCH JOURNAL OF TECHNOLOGY AND APPLIED SCIENCE
http://www.irjtas.com
(An ISO Certified Journal)
VOL. 05 Issue 02 FEBRUARY 2021

```
model = cnn_model()
```

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 32, 32, 32)	896
conv2d_2 (Conv2D)	(None, 30, 30, 32)	9248
max_pooling2d_1 (MaxPooling2	(None, 15, 15, 32)	0
dropout_1 (Dropout)	(None, 15, 15, 32)	0
conv2d_3 (Conv2D)	(None, 15, 15, 64)	18496
conv2d_4 (Conv2D)	(None, 13, 13, 64)	36928
max_pooling2d_2 (MaxPooling2	(None, 6, 6, 64)	0
dropout_2 (Dropout)	(None, 6, 6, 64)	0
flatten_1 (Flatten)	(None, 2304)	0
dense_1 (Dense)	(None, 512)	1180160
dropout_3 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 10)	5130

```
Total params: 1,250,858
Trainable params: 1,250,858
Non-trainable params: 0
```

Figure 4. Simple CNN model building

Afterwards, the model is trained on the training data. ―ModelCheckpoint()‖ method saves the simplest model after every epoch. Lastly, the model fits the provided data employing a batch size adequate to 128, which is that the number of samples per gradient update. The model uses 100 iterations (epochs).

After the successful training, the model is evaluated and presents the accuracy and loss on the matplotlib graphs. The IPython notebook includes few scripts for predicting class for the test set of images shown in figure 5.

Predict class for test set images

```
class_pred = model.predict(images_test, batch_size=32)
print(class_pred[0])
```

```
[  2.34778727e-05   3.61718005e-03   7.07750034e-04   6.86623812e-01
   2.35363492e-04   2.97410578e-01   9.98710049e-04   1.62866409e-03
   8.46819486e-03   2.86295195e-04]
```

Get the index of the largest element in each vector

```
labels_pred = np.argmax(class_pred,axis=1)
print(labels_pred)
```

```
[3 8 8 ..., 5 1 7]
```

Check which labels have been predicted correctly

```
correct = (labels_pred == labels_test)
print(correct)
print("Number of correct predictions: %d" % sum(correct))
```

```
[ True  True  True ...,  True  True  True]
Number of correct predictions: 7883
```

Calculate accuracy using manual calculation

```
num_images = len(correct)
print("Accuracy: %.2f%%" % ((sum(correct)*100)/num_images))
```

```
Accuracy: 78.83%
```

Figure 5 Sample predictions

Improved CNN Model

Firstly, the notebook requires importing a couple of additional functions from Keras and decoder packages. the method of importing class names also as fetching and decoding the info remains unchanged.

Definition of the improved CNN model consists of the foremost essential changes shown in figure 6. the improved model replaces the max-pooling and dense function with two-dimensional convolution layers. The architecture uses nine layers with a special number of convolution filters. within the end, the model uses the operation of two-dimensional global average pooling. After the structure definition, the model is made and summarized.

.

ISSN (online):2396-9889
INTERNATIONAL RESEARCH JOURNAL OF TECHNOLOGY AND APPLIED SCIENCE
http://www.irjtas.com
(An ISO Certified Journal)
VOL. 05 Issue 02 FEBRUARY 2021

```
model = improved_cnn_model()
```

Layer (type)	Output Shape	Param #
conv2d_10 (Conv2D)	(None, 32, 32, 96)	2688
dropout_4 (Dropout)	(None, 32, 32, 96)	0
conv2d_11 (Conv2D)	(None, 32, 32, 96)	83040
conv2d_12 (Conv2D)	(None, 16, 16, 96)	83040
dropout_5 (Dropout)	(None, 16, 16, 96)	0
conv2d_13 (Conv2D)	(None, 16, 16, 192)	166080
conv2d_14 (Conv2D)	(None, 16, 16, 192)	331968
conv2d_15 (Conv2D)	(None, 8, 8, 192)	331968
dropout_6 (Dropout)	(None, 8, 8, 192)	0
conv2d_16 (Conv2D)	(None, 8, 8, 192)	331968
activation_4 (Activation)	(None, 8, 8, 192)	0
conv2d_17 (Conv2D)	(None, 8, 8, 192)	37056
activation_5 (Activation)	(None, 8, 8, 192)	0
conv2d_18 (Conv2D)	(None, 8, 8, 10)	1930
global_average_pooling2d_2 (	(None, 10)	0
activation_6 (Activation)	(None, 10)	0

```
Total params: 1,369,738
Trainable params: 1,369,738
Non-trainable params: 0
```

Figure 6. Improved CNN model building

The model is trained using the same parameters, where the only difference is the increased number of 350 epochs. and the prediction result are shown in figure 7.

Evaluate the model

```
scores = model.evaluate(images_test, class_test, verbose=0)
print("Accuracy: %.2f%%" % (scores[1]*100))
```

```
Accuracy: 84.96%
```

Figure 7. improved Model prediction

Comparison

The accuracy, as well as running time of all the tested models, are presented in the following table.

Table 1. Accuracy Of The CNN Models

Classification Model	Accuracy
Simple CNN	78.83%
Modified CNN	84.96%

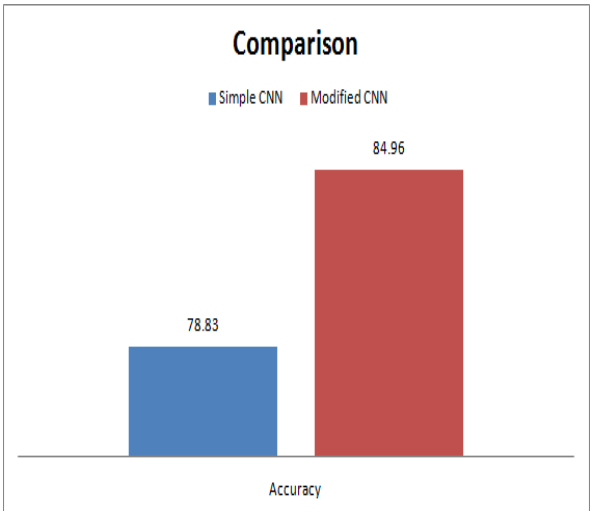

Figure 8. Comparison of models

The results shows orverall 10% improvement between the Traditional and modified model utilized in the experiment. CNN model after dropping max-pooling and dense function did improve its accuracy up to 87.94%; however, as of running 3 times more epochs, the time period increased for the improved CNN.

VI CONCLUSION

Neural networks and deep learning is a neighborhood of research with tons of unanswered questions, a neighborhood with such intensive research that it are often difficult to stay pace with all the new findings that are discovered constantly. A practitioner got to be conversant in the underlying theory to ready to work with neural networks and deep learning and also got to be familiar with their building blocks to be ready to

ISSN (online):2396-9889
INTERNATIONAL RESEARCH JOURNAL OF TECHNOLOGY AND APPLIED SCIENCE
http://www.irjtas.com
(An ISO Certified Journal)
VOL. 05 Issue 02 FEBRUARY 2021

use them effectively. a really important thing to stress here that within the process of developing machine learning applications the standard of the available data, not just the standard of the code, affects the top quality of the merchandise , unlike other areas of software development. The performance of a machine learning model can nearly always be improved with more and better data. .In this dissertation we have taken convolution neural network with different layers configuration and apply on CIFAR-10 dataset and we found that our modified CNN model takes more time but perform better as compared to Traditional CNN.

REFERENCES

[01] Yifeng Zhao1 ,Weimin Lang1 ,Bin Li2 ―Performence analysis of neural network with improved weight training process‖ in IEEE 2019.

[02] Raniah Zaheer , Humera Shaziya ―A Study of the Optimization Algorithms in Deep Learning‖ in IEEE 2019.

[03] Sara Mourad, Haris Vikalo and Ahmed Tewfik ―ONLINE SELECTIVE TRAINING FOR FASTER NEURAL NETWORK LEARNING‖ in IEEE 2019.

[4] K. Simonyan and A. Zisserman, ―Very deep convolutional networks for Large-Scale image recognition,‖ Sep. 2014.

[5] S. Liu and W. Deng, ―Very deep convolutional neural network based image classification using small training sample size,‖ in 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR), Nov. 2015, pp. 730–734.

[6] Y. Chen, Y. Yang, W. Wang, and C. C. Jay Kuo, ―Ensembles of feedforward-designed convolutional neural networks,‖ Jan. 2019.

[7] Y. Huang, Y. Cheng, A. Bapna, O. Firat, M. X. Chen, D. Chen, H. Lee, J. Ngiam, Q. V. Le, Y. Wu, and Z. Chen, ―GPipe: Efficient training of giant neural networks using pipeline parallelism,‖ Nov. 2018.

[8] M. Tan and Q. V. Le, ―EfficientNet: Rethinking model scaling for convolutional neural networks,‖ May 2019.

[9] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, ―AutoAugment: Learning augmentation policies from data,‖ May 2018.

[10] N. Nayman, A. Noy, T. Ridnik, I. Friedman, R. Jin, and L. Zelnik-Manor, ―XNAS: Neural architecture search with expert advice,‖ Jun. 2019.

[11] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, ―ImageNet: A large-scale hierarchical image database,‖ in 2009 IEEE Conference on Computer Vision and Pattern Recognition, Jun. 2009, pp. 248–255.

[12] IPython Development Team (2018) Ipython Interactive Computing [online] available from <https://ipython.org/> [26 April 2018]

[13] Project Jupyter (2018) Jupyter [online] available from <http://jupyter.org/> [26 April 2018]

[14] Chollet, F. (2018) Keras Documentation [online] available from <https://keras.io/> [26 April 2018]

[15] Google Brain Team (2018) Tensor flow [online] available from <https://www.tensorflow.org/> [21 April 2018]

[16]. F. J. Huang and Y. LeCun, "Large-scale learning with SVM and convolutional nets for generic object categorization," in Proceedings of the IEEE Computer Society Conference on

ISSN (online):2396-9889
INTERNATIONAL RESEARCH JOURNAL OF TECHNOLOGY AND APPLIED SCIENCE
http://www.irjtas.com
(An ISO Certified Journal)
VOL. 05 Issue 02 FEBRUARY 2021

Computer Vision and Pattern Recognition (CVPR '06), pp. 284–291, New York, NY, USA, June 2006.

[17]. X.-X. Niu and C. Y. Suen, "A novel hybrid CNN-SVM classifier for recognizing handwritten digits," Pattern Recognition, vol. 45, no. 4, pp. 1318–1325, 2012.

[18].S. Bianco, M. Buzzelli, D. Mazzini, and R. Schettini, "Logo recognition using CNN features," in Image Analysis and Processing—ICIAP 2015, vol. 9280 of Lecture Notes in Computer Science, pp. 438–448, Springer, Cham, Switzerland, 2015.

[19].S. Wang and J. Lai, "A more complex neuron in biomimetic pattern recognition," in Proceedings of the International Conference on Neural Networks and Brain Proceedings (ICNNB '05), vol. 3, Beijing, China, October 2005.

[20]A. D. Aleksandrov, Mathematics, Its Essence, Methods and Role, USSR Academy of Sciences, Moscow, Russia, 1956.

[21]K. Fukushima, "Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," Biological Cybernetics, vol. 36, no. 4, pp. 193–202, 1980.

[22]Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.

[23]P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis," in Proceedings of the 7th International Conference on Document Analysis and Recognition (ICDAR '03), pp. 958–963, August 2003.

1. [24]D. C. Cireşan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, "Flexible, high performance convolutional neural networks for image classification," in Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI '11), vol. 22, pp. 1237–1242, July 2011.

2. [25]D. H. Hubel and T. N. Wiesel, "Receptive fields and functional architecture of monkey striate cortex," The Journal of Physiology, vol. 195, no. 1, pp. 215–243, 1968.

3. [26]D. Scherer, A. Müller, and S. Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition," in Proceedings of the International Conference on [27]Artificial Neural Networks (ICANN '10), vol. 6354, no. 3, pp. 92–101, Thessaloniki, Greece, September 2011. Y. Lecun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," Neural Networks: Tricks of the Trade, vol. 7700, pp. 9–48, 2012.

4. [28]D. Strigl, K. Kofler, and S. Podlipnig, "Performance and scalability of GPU-based convolutional neural networks," in Proceedings of the 18th Euromicro Conference on Parallel, Distributed and Network-Based Processing (PDP '10), pp. 317–324, Pisa, Italy, February 2010.

5. [29] Y. Ren, Y. Wu, and Y. Ge, "A co-training algorithm for EEG classification with biomimetic pattern recognition and sparse representation," Neurocomputing, vol. 137, pp. 212–222, 2014.

6. [30]J.-J. Seo, H.-I. Kim, and Y. M. Ro, "Pose-robust and discriminative feature representation by multi-task deep learning for multi-view face recognition," in Proceedings of the 17th IEEE International Symposium on Multimedia (ISM '15), pp. 166–171, Miami, Fla, USA, December 2015.

7. [31]G. Jingyu, J. Yang, J. Zhang, and M. Li, "Natural scene recognition based on Convolutional Neural Networks and Deep Boltzmannn Machines," in Proceedings of the IEEE International Conference on Mechatronics and Automation (ICMA '15), vol. 2015, pp. 2369–2374, 2015.

8. [32]Y. Zhai, J. Li, J. Gan, and Y. Xu, "A novel SAR image recognition algorithm with rejection mode via biomimetic pattern recognition," Journal of Information and Computational Science, vol. 10, no. 11, pp. 3363–3374, 2013.

[33] W. Cao, H. Feng, L. Hu, and T. He, "Space target recognition based on biomimetic pattern recognition," in Proceedings of the 1st International Workshop on Database Technology and Applications (DBTA '09), April 2009.

ISSN (online):2396-9889
INTERNATIONAL RESEARCH JOURNAL OF TECHNOLOGY AND APPLIED SCIENCE
http://www.irjtas.com
(An ISO Certified Journal)
VOL. 05 Issue 02 FEBRUARY 2021

[34]J.-Y. Zeng, J.-Y. Gan, and Y.-K. Zhai, "A novel partially occluded face recognition method based on biomimetic pattern recognition," in Proceedings of the International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR '12), pp. 175–179, Xian, China, July 2012.

[35]C. Yao, F. Wu, H.-J. Chen, X.-L. Hao, and Y. Shen, "Traffic sign recognition using HOG-SVM and grid search," in Proceedings of the 12th IEEE International Conference on Signal Processing (ICSP '14), pp. 962–965, IEEE, Hangzhou, China, October 2014.

[36]The MNIST database, http://yann.lecun.com/exdb/mnist/index.html.

[37]"The AR face database," http://www2.ece.ohio-state.edu/~aleix/ARdatabase.html.

[38]"The CIFAR-10 dataset," http://www.cs.utoronto.ca/~kriz/cifar.html.