International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 06 | June 2021 www.irjet.net p-ISSN: 2395-0072

OBJECT DETECTION USING CIFAR-10 DATASET SIMPLENET

Shakti Punj[1], Aishwarya Punj[2], Ambika Punj[3]

[1]*Department of CSE, SRM Institute of Science and Technology, Modinagar, Delhi NCR*
[2,3]*Department of Mathematics, G.D. Goenka Public School, Agra, UP*

Abstract - *Convolutional Neural Network often known as ConvNet is the algorithm used here; it has an advance segment called Max-pooling which handles the details. It has self-learning Neuron which takes many inputs and pass outputs accordingly.*

CNN constitutes several layers of perceptrons. It has several neurons in each layer designed such to avoid over fitting of data. It requires lesser pre-processing unlike others, unlike other algorithms where data is to be stored synthesized manually here data is self-learn and taught.

Using these principles, we see that SimpleNet provides best results and a platform for this self-learning visual mechanism. To provide best result the matrix containing epochs are to be varied. Henceforth cifar-10 data set is provided in this algorithm designed platform and then machine is trained accordingly.

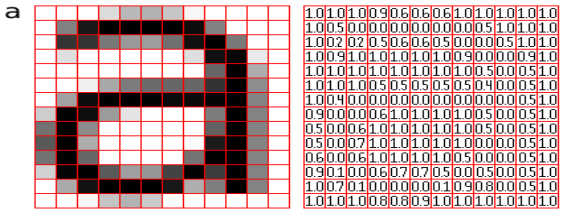
Using CNN it resolves problems like overfitting and also reduces time complexity. Its can have input as images in proper grayscale size and then the images are fragmented in smaller sizes for better clearity and understanding of machine.

Cifar-10 has 10 distinct classes which are mutually exclusive and these classes. It has total 60,000 images, of which 50,000 are used for training and rest for testing. Testing has ways, with data augmentation or without. Altering inputs and changing epochs and its input matrix we can obtain better result.

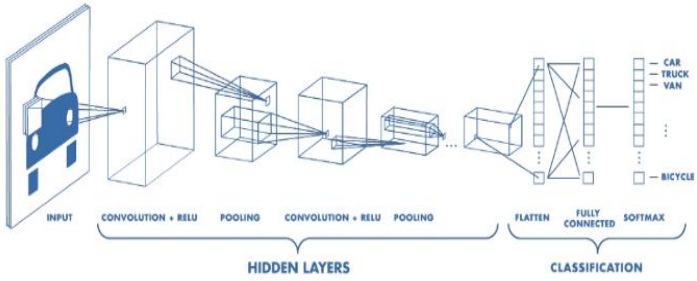
Key Words: Object Detection, Simplenet, CNN, Image Recognition and Identification.

INTRODUCTION

Convolutional Neural Network (CNN)CNN is a special way of converting image into a grid like mechanism also known as ConvNet. Each pixel so formed is unique to illustrate the color-combination and bright-ness of each area of image.



CNN Architecture- It is a three layered system namely-• convolutional • pooling • fully connected



1st Layer (Convolution) 2 matrices specifically 1 with learnable parameters(kernels) and other matrix which is in the restricted field of receptive field are dot produced. Its height and width will be relatively smaller than its depth (can go up to 3 units) Pooling Operation. Pooling layer this layer reduces certain values from visual field which reduces size and time complexity of the algorithm thus making it light weight and output is generated overcoming over fitting. Most widely used way is maxpooling in which output is generated using neighbor values. Object detection using Cifar-10 SimpleNet technology includes a very light weight architecture, Cifar10 is a data set through which we train our machine, first of all data is being understood and then converted to computer algorithm and fed to the machine for recognition. Cifar-10The Cifar-10 data set has 60,000 images of which comprises of 50000 images to train and 10,000 images to test. Among those 10,000 images 1000 images are randomly selected from each class Figure 1.4: model1.4 design DataSet are designed such that none of the classes overlap each other. Matrix for each entered data is formed and searched. SimpleNet is the framework written in java to receive and sent requested data/information.

KEY FEATURES - In this paper we propose how to make training, testing searching efficient. Memory consumption can be drastically reduced using Deep Compression. Here we explain objectives, architecture its inputs, results and efficiency and has related data to support the argument. An efficient system has less number of layers and is as good as or better than its deep long version. Image is taken by camera or by mobile and is provided as input. Image Pre-processing- After the input is given image goes through several steps which are hidden and is fragmented to several pieces for better understanding. When image is processed with digital processing and algorithms it has various advantages over images processed by analog4

Image processing – After the pre-processing is done separate mathematical matrix are produced which helps in key point detection, a special filter array know as kernels is formed in that domain. Due to this complex methodology it requires special algorithms. Key points detection the maximized image is then properly studied and various things like what are the edges, what is shape color and size of specific parts are observed similar to the pattern recognition feature in machine learning. The image so obtained with specific extracted data is called feature image. When the related sub details are fetched it needs to be stored, therefore database descriptors stores the related data. Several entity and attribute class are designed to sequentially arrange data from various tables.

IF NEW INTEREST REGION IS FORMED - It is added as a new descriptor if no the nit goes to KEY POINT MATCHING where related output is generated.

RELATED WORK

Sharpness - Aware Minimization for Efficiently Improving Generalization

general idea Pierre Foret • Ariel Kleiner • Hossein Mobahi • Behnam Neyshabur in 2020 proposed the idea of how sharpness of image is done. Since training is to be done of a very big dataset so how to overcome over fitting is a big task

Evaluating empirically

In order to improve efficacy proper trial and error was done than relying of theoretical values through which SAMs efficacy was improved using dataset Cifar10/100 and ImageNet.

An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale.}

Need

Alexey Dosovitskiy • Lucas Beyer • Alexander Kolesnikov • Dirk Weissenborn • Xiaohua Zhai • Thomas Unterthiner • Mostafa Dehghani in 2020 stated that transformer and image fragmentation is used to transfer large training data to smaller parts like ImageNet and VTAB to provide good results

Ideology

When a transformer is applied to the image fragments either some of the components are replaced or removed, here it is stated that we need not depend on CNN, we can make changes according to the requirements

Big Transfer (BiT): Visual Learning representation

Need

Alexander Kolesnikov • Lucas Beyer • Xiaohua Zhai • Joan Puigcerver in 2019 stated using combination of smaller simpler elements with help of heuristic on 20 dataset effective results can be achieved.

Big Transfer

Here we use two types of streaming upstream which has data for preprocessing and downstream is used for fine tuning of new task. We create 3 BiTs for data processing and real world dataset is interconnected through object detector.

Incorporating Convolution Designs into Visual Transformers

Need

Kun Yuan • Shaopeng Guo • Ziwei Liu • Aojun Zhou in 2021 stated in order to apply vision domain on BiT and DeiT. We change architecture to obtain data as it requires large number of dataset and training time. Unlike other ways tokens are directly joined to images which generate low level features.

Locally-Enhanced Feed-Forward Network

We split the normal tokens into patch tokens and perform deep convolutional neural networks. Here importance of virtual dataset and training methods are discussed, dataset being JFT300M or ImageNet22K.

Training data-efficient image transformers & distillation through attention

Need

Hugo Touvron • Matthieu Cord • Matthijs Douze • Francisco Massa in 2020 stated using large infrastructure and input data images can be fed to training and testing dataset on a single desktop data was trained in less than three days. We do not need external tokens or dataset and still we can achieve good results

Knowledge about distillation there is a stable and good teacher network and small student network with soft labels. These kind of student network is subset of teacher module and thus has very reliable results

Sample-Efficient Neural Architecture Search by Learning Action Space for Monte Carlo Tree Search

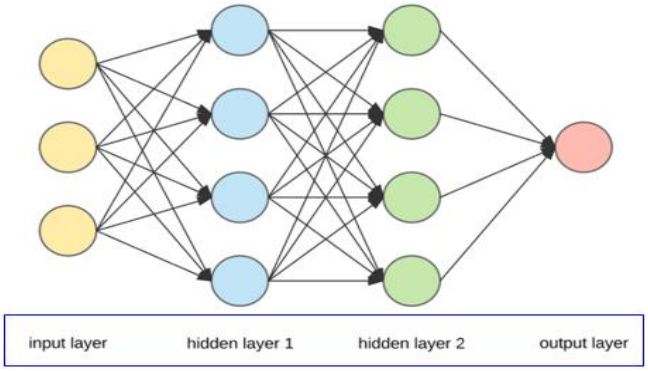
Need

Linnan Wang, Saining Xie, Teng Li, Rodrigo Fonseca in 2019 stated that NAS has a advance way of automatic convolutional due to which a inefficient data is produced therefore a better and stable version called latent action neural architecture research is proposed.

Motivating and Learning

Both one shote and LaNAS provide short as well as long term results by biasing towards good regions, Good from bad models is extracted to provide the resulting data for training. Empirical result has shown that LaNAS has shown effective results.

SYSTEM ANALYSIS



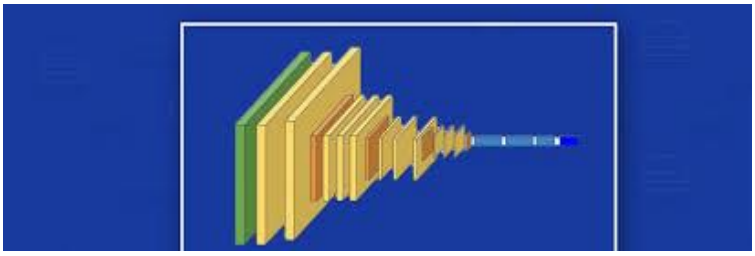
Several important aspects of CNN is elaborated and explained with support of relative validate data with comparison based on experimental outputs reference to such postulates it compares the previously used technologies where self-learning was not an option and furthermore states why it is relatively more better and widely accepted.

It further compares its architecture with other pre-existing best practices strided vs max-pooling and overlapped vs non overlapped pooling It has various sub segments such as MS-CNN which makes searching recognizing easier hence faster. It reduces the memory costing and increases efficiency. has R-CNN which without rearranging or managing data techniques is able to scale thousands of object and its classes, it does not involve primitively used selective search algo-techniques it instead uses a different mechanism to learn and understand object/image

ARCHITECTURAL DESIGN FOR PROPOSED SYSTEM

It has two different layers of convolutional and average pooling respectively. And then it is joined to a flattening convolutional layer followed by two connected layers and ending at a softmax classifier.

I Layer - Images are taken at grid specification of 32×32 and then it passes through a filter of size 5×5 which is in first convolutional layer and hence the image dimension is altered.



II Layer - A stride with 2 of size 2×2 is applied in pooling or sub-sampling layer resulting in reduction of image size to 14×14×6.

III layer - This layer has sixteen features and lies in 2nd convolutional layer, 10 features are connected to 6 of the previous one. This is the technique used to make it non symmetric and not to exceed the bound range training parameter now - 1516, previous -2400 connections now - 151600, previous – 240000.

IV Layer - It is similar to second layer but it has 16 features so the expected result is de-creased to 5×5×16.5.1.5

V Layer - Constituents are size 1×1 and has 120 features every u in it is interconnected to400 nodes

VI Layer - has all the units connect i.e. 84.

Output – Now system is fully connected with 10 outputs ranges 0-9 digits.

METHODOLOGY

It functions upon a 60000 colored photographs of size 32×32. Since we are discussing cifar-10 dataset it has 10 classes namely-• airplane• automobile• bird• cat• deer•

dog• frog• horse• ship• truck Dataset formed is for a typical machine like computer whereas it has minute size pictures which are much smaller than an actual picture. As neural networking is used image identification and classification is a tough task so this has been included by various techniques to improve accuracy clarity and identifying. algorithm and dataset - CIFAR-10 is a widely accepted and commonly used dataset due to its distinct features. To achieve a good accuracy it is practically impossible without it.Cifar-10 dataset proved to be beneficial in achieving accuracy by deep convolutional technique using neural network. Here is an example in which keras API issued and 9 picture dataset is created from training set.

Image fragmentation description- Here is a description of how image fragmentation is done and aligned for the detection of location, number of objects and type of object

INPUT MATRIX - using separate array called kernals a special dataset for image is formed and then it is altered and various data augmentation steps which are as follows -

IMAGE CLASSIFICATION- image is classified into smaller grid size, so that specifically each point can be examined. Whether image is blurring, discrete and how many objects are there.

OBJECTS LOCALISATION -It involves determining location in form of grid size its edges etc.

OBJECTS DETECTION - Finally object is detected and number of objects and there type is known

```
# example of loading the cifar10 dataset
from matplotlib import pyplot
from keras.datasets import cifar10
# load dataset
(trainX, trainy), (testX, testy) = cifar10.load_data()
# summarize loaded dataset
print('Train: X=%s, y=%s' % (trainX.shape, trainy.shape
print('Test: X=%s, y=%s' % (testX.shape, testy.shape))
# plot first few images
for i in range(9):
    # define subplot
    pyplot.subplot(330 + 1 + i)
    # plot raw pixel data
    pyplot.imshow(trainX[i])
# show the figure
pyplot.show()
```

An array matrix is created for 50000 images from training set and testing set has 10000. Image size being 32 by 32 which are colored consisting of 3 channels.

```
Train: X=(50000, 32, 32, 3), y=(50000, 1)
Test: X=(10000, 32, 32, 3), y=(10000, 1)
```

ALGORITHMS AND TECHNIQUES USED

CIFAR10/100 - this dataset includes 60,000 colored images. Its data is divided into 10/100 mutually exclusive classes. Here we have taken two sample inputs one using augmented data and one without using it i.e..

CIFAR10/100 - this dataset includes 60,000 colored images. Its data is divided into 10/100 mutually exclusive classes. Here we have taken two sample inputs one using augmented data and one without using it i.e.. We state them Exp1 and Exp2 respectively. We see entirely different scenario in both Exp1 and Exp2. With data augmentation we received 76 percent accuracy

The MNIST dataset - has total of 70,000 images with grid of 28x28, they have handwritten digits,70,000 images comprises of 60,000 and 10,000 which are for training and respectively . Data augmentation was not used here still its accuracy was satisfactory.

The SVHN dataset - is the widely used dataset which has real world inputs. It has a total of 630,420 32x32 colored images. Training set has 73,257 images, testing set has 26,032 and remaining 531,131 images are used for additional training. Like previously referred data we used training and testing dataset and no data augmentation was done.

RESULT

```
1563/1563 [==============================] - 33s 21ms/step - loss: 0.7812 - accuracy: 0.7396 - val_loss: 0.6617 - val_accuracy: 0.7794
Epoch 91/100
1563/1563 [==============================] - 32s 21ms/step - loss: 0.7789 - accuracy: 0.7434 - val_loss: 0.7390 - val_accuracy: 0.7510
Epoch 92/100
1563/1563 [==============================] - 33s 21ms/step - loss: 0.7902 - accuracy: 0.7374 - val_loss: 0.6845 - val_accuracy: 0.7696
Epoch 93/100
1563/1563 [==============================] - 33s 21ms/step - loss: 0.7986 - accuracy: 0.7348 - val_loss: 0.6708 - val_accuracy: 0.7757
Epoch 94/100
1563/1563 [==============================] - 33s 21ms/step - loss: 0.8045 - accuracy: 0.7343 - val_loss: 0.7392 - val_accuracy: 0.7526
Epoch 95/100
1563/1563 [==============================] - 32s 21ms/step - loss: 0.7983 - accuracy: 0.7348 - val_loss: 0.7351 - val_accuracy: 0.7501
Epoch 96/100
1563/1563 [==============================] - 33s 21ms/step - loss: 0.8064 - accuracy: 0.7322 - val_loss: 0.7133 - val_accuracy: 0.7621
Epoch 97/100
1563/1563 [==============================] - 33s 21ms/step - loss: 0.7899 - accuracy: 0.7399 - val_loss: 0.7487 - val_accuracy: 0.7659
Epoch 98/100
1563/1563 [==============================] - 32s 20ms/step - loss: 0.8016 - accuracy: 0.7360 - val_loss: 0.7012 - val_accuracy: 0.7753
Epoch 99/100
1563/1563 [==============================] - 32s 21ms/step - loss: 0.8014 - accuracy: 0.7349 - val_loss: 0.7584 - val_accuracy: 0.7539
Epoch 100/100
1563/1563 [==============================] - 32s 21ms/step - loss: 0.8028 - accuracy: 0.7353 - val_loss: 0.6939 - val_accuracy: 0.7690
Saved trained model at /content/saved_models/keras_cifar10_trained_model.h5
313/313 [==============================] - 1s 3ms/step - loss: 0.6939 - accuracy: 0.7690
Test loss: 0.6939066052436829
Test accuracy: 0.7689999938011169
```

FUTURE WORK

FUTURE ENHANCEMENT Initially we used primitive ways of image recognition but its efficiency was very less. It was not a robust approach also its time complexity was very less and it required several extra steps and datasets. Then we designed a proper schematic way in which we altered epochs and got a sustainable result. Accuracy rate of 76 percent was obtained. We got amazing results and in future we will add GPU and will train data accordingly. In real world just recognizing shapes and images is not enough, we need a proper mechanism which can detect multiple objects or blur objects. The algorithm and technique designed by us has functionality to understand multiple images. Since it is machine learning based technique it often becomes tough to

tell what is object and what should not be considered its object but just side things say for example there is an image of a plant, all the constituents. Its branches and leaves are still considered a single plant. But if the same leaves would have been lying around it will be considered different objects. Neural network and perceptron framework finds it difficult to understand what an object is and what its background

CONCLUSIONS

Going through the whole projects we were able to learn several tips and tricks and how we can alter image dataset object grid size so we hereby conclude that by learning proper algorithms there input matrix there techniques we can obtain best results. We also understand some key features means an algorithm in object detection must be

Light weight - for better and fast results the input and requirements must be light weight and should have light weight architecture. Image searching should be done by similar fragments and must not require various layers.

Fast speed - search and detection engines must work fast so as to reduce the time involved in frag-mentation, comparison or while delivering results. Accuracy with speed is best combination which can be obtained by altering the matrix and changing number of epochs

Must function in Deformed images - whether the images are of good quality or deformed the detection process should be equally good and must follow similar principles. maxpooling CNN layers contain several pixel enlarging tools which are used to for the same reason

Less Equipment - It should not involve many tools and should be cost efficient. It has some hidden layers which also carries weight. Weight of respective classes is calculating by multiplying input by its corresponding weight and then adding to provide the single result.

REFRENCES

[1] Pierre Foret • Ariel Kleiner.2020. Sharpness-Aware Minimization for Efficiently Improving Generalization. EffNet-L2 (SAM)

[2] Alexey Dosovitskiy • Lucas Beyer 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. ViT-H/14.

[3] Alexey Dosovitskiy • Lucas Beye.2021 An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. ViT-L/16.

[4] Hugo Touvron • Matthieu Cord.2021. Going deeper with Image Transformers CaiT-M-36 U 224.

[5] Haiping Wu • Bin Xiao.2021 CvT Introducing Convolutions to Vision Transformers CvT-W24

[6] Alexander Kolesnikov • Lucas Beyer • Xiaohua Zha.2019 Big Transfer (BiT): General Visual Representation Learning. BiT-L(ResNet).

[7] Kun Yuan • Shaopeng Guo • Ziwei Liu.2021 Incorporating Convolution Designs into Visual Transformers CeiT-S (384 finetune resolution).

[8] Kai Han • An Xiao • Enhua WuJianyuan Guo • Chunjing Xu • Yunhe Wang.2021 Transformer in Transformer TNT-B.

[9] Hugo Touvron • Matthieu Cord • Matthijs Douze • Francisco Massa.2020 Training data-efficient image transformers & distillation through attention DeiT-B.

[10] Mingxing Tan • Quoc V. Le,2021 EfficientNetV2: Smaller Models and Faster Training EfficientNetV2-L.

[11] Linnan Wang • Saining Xie • Teng Li • Rodrigo Fonseca.2019 Sample-Efficient Neural Architecture Search by Learning Action Space for Monte Carlo Tree Search LaNe.

[12] Sungbin Lim • Ildoo Kim • Taesup Kim • Chiheon Kim • Sungwoong Kim.2019 Fast AutoAugment PyramidNet+ShakeDrop (Fast AA)