Malnad College of Engineering

(An Autonomous Institution under Visvesvaraya Technological University, Belagavi)

Hassan – 573 202

A MINI PROJECT REPORT

ON

"Recognition of Objects with Convolutional Neural Network for Automated Image Organization"

Submitted by,

Name: Hemaraj H N	**USN: 4MC18CS051**
Jayshankar	**4MC18CS057**

Under the guidance of

Dr. Ramesh B

Senior Professor

Department of Computer Science and Engineering

Malnad College of Engineering

Hassan - 573 202

JUNE - 2021

Malnad College of Engineering

(An autonomous Institution under Visvesvaraya Technological University, Belagavi)

Hassan – 573 202

Department of Computer Science and Engineering

CERTIFICATE

Certified that the Mini Project Work titled "Online Shopping Website"
presented by

Name: Hemaraj H N	**USN: 4MC18CS051**
Jayshankar	**4MC18CS057**

during the year 2020-2021 is satisfactory and is approved as a creditable engineering study in the 6th semester. The work is accepted as a prerequisite for the award of degree of Computer Science and Engineering.

Dr.Ramesh B Prof., Dept of CSE	Dr. Geetha Kiran A B.E., M.Sc (Engg) by Research.,Ph.D., **Head of the Department**	Dr. C.V Venkatesh B.E.,M.Tech.,Ph.D., **Principal**

External Viva

Name of the Examiners	**Signature with Date**
1.	
2.	

ACKNOLEDGEMENT

We sincerely express our gratitude to our guide "Dr. Ramesh" Dept. of Computer Science and Engineering, Malnad College of Engineering, Hassan for his precious guidance and help in this mini- project work. This credit for the successful completion of this mini-project goes to his keen interest timing guidance and valuable suggestion otherwise our endeavor would have been futile.

We sincerely thank to **" Dr. Ramesh B, Prof. Harshitha S and Prof. Vinodh A M"** for their guidance and encouragement in the completion of this mini-project.

We owe to regard to **" Dr. Geetha Kiran A "** Head of the department of Computer Science and Engineering for her persistent encouragement and blessing which were bestowed upon us.

ABSTRACT

The Convolutional Neural Networks have become the most powerful method for image classification. Image classification is the process of segmenting images into different categories based on their features. Here the network identifies the features like edges in an image, the pixel intensity, the change in pixel values, and many more with the help of pre-processing techniques. The best possible outcome can be obtained in terms of accuracy by using Convolutional Neural Network.

Contents

CHAPTER 01

1.1 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 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.

A convolutional neural network (CNN) is a type of artificial neural network used in image recognition and processing that is specifically designed to process pixel data.

1.2 Objective

The aim of this project is to build a deep learning model so as to recognize the object by using CIFAR-10 data set (Canadian Institute for Advanced Research). To build a deep learning model so as to recognize the object by using CIFAR-10 data. To get the best possible outcome in the prediction of images in terms of accuracy.

The classification problem is to categorize all the pixels of a digital image into one of the defined classes. Image classification is the most critical use case in digital image analysis. Image classification is an application of both supervised classification and unsupervised classification.

- In supervised classification, we select samples for each target class. We train our neural network on these target class samples and then classify new samples.
- In unsupervised classification, we group the sample images into clusters of images having similar properties. Then, we classify each cluster into our intended classes.

CHAPTER 02

CASE STUDY:

2.1 Literature Survey

The traditional convolutional neural network usually initializes the weights of all network layers at one time before network training, and then updates the weights of the network by back-propagation algorithm to improve the accuracy of the network during network training. However, with the increase of network depth, the computational cost of this method will increase dramatically and the test accuracy will be affected. In order to solve this problem, a method of gradually reinitializing the weights of each layer is proposed, that is, after a certain training period, the weight of the previous layer is determined and remain unchanged, then initialize the weights of all subsequent layers, repeat this step until the weights of all layers are determined. In order to verify the performance of the method, a series of experiments were carried out on the CIFAR10 dataset. The results show that the accuracy of the network is improved by 9% and the training time is reduced by 29%. It shows that the method can improve the accuracy of the network and reduce the training time.

2.2 Object Recognition:

Object recognition is an important task, yet challenging vision task. It is a critical part of many applications such as image search, image auto-annotation and scene understanding, object tracking. Moving object tracking of video image sequences was one of the most important subjects in computer vision. It had already been applied in many computer vision fields, such as smart video surveillance, artificial intelligence, military guidance, safety detection and robot navigation, medical and biological application.

CHAPTER 03

SYSTEM SPECIFICATION AND TOOLS

3.1 Existing system:

- ❑ Currently, ANN- Artificial Neural Network is used for image classification. The main disadvantage of ANN is that it is Hardware dependent, its unexplained behavior of the network.
- ❑ The difficulty of showing the problem to the network: Problems have to be translated into numerical values before being introduced to ANN.
- ❑ And also, ANN has less accuracy than that of Convolutional Neural Network.

3.2 Methodology:

❑ To overcome the demerits of image recognition using ANN, we can use Convolutional Neural Network.

❑ Convolutional Neural Network gives the best possible accuracy in recognizing the images compared to ANN.

❑ CNN can directly analyze images, there is no need of translating the images into numerical data.

3.3 Hardware Requirements: We don't need a big system. We could even skip the GPUs altogether. A CPU such as i7–7500U can train an average of ~115 examples/second is enough.

System: I5
Hard Disk: 1 TB.
Ram: 4GB.

3.4 Software Requirements:

Operating system: Windows 10

Coding Language: PYTHON

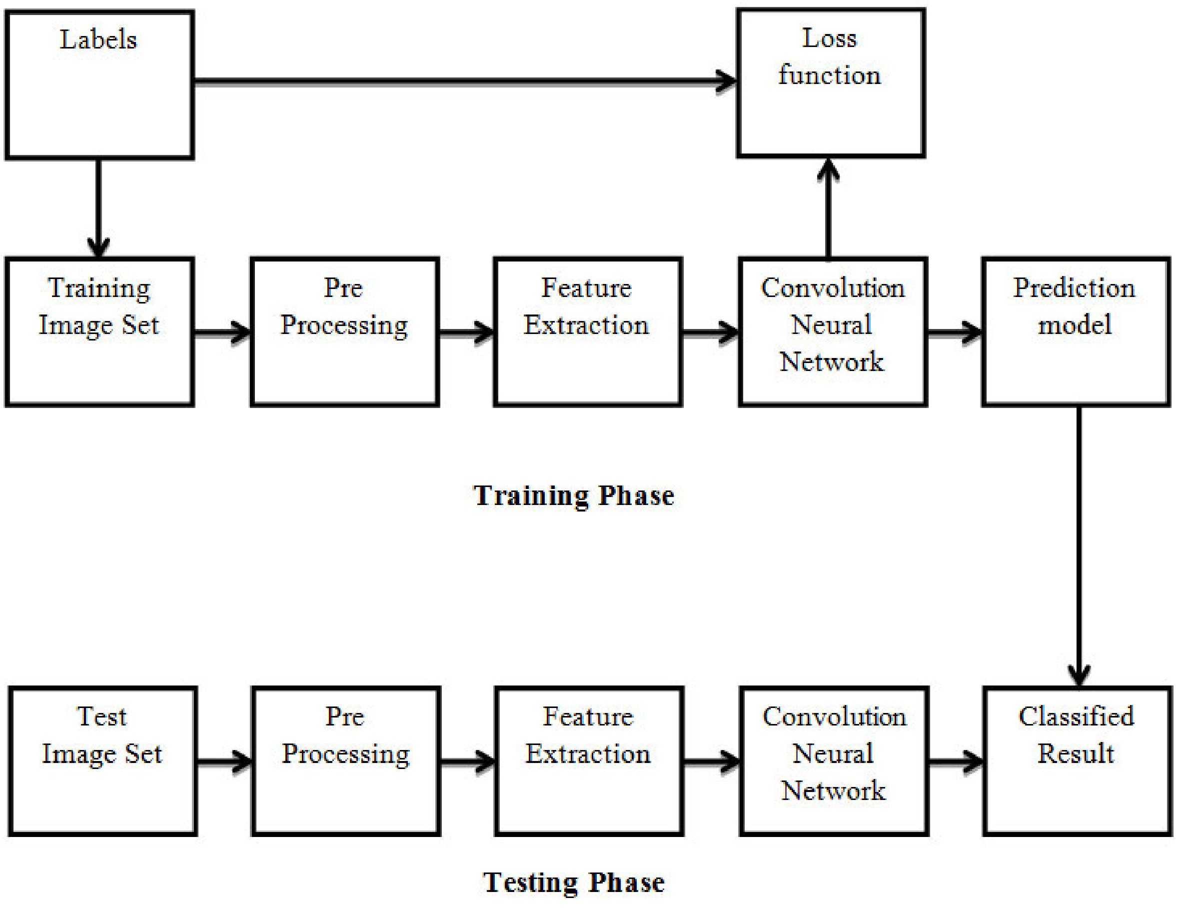
Editor: Google Collaboratory

CHAPTER 04

SOFTWARE DESIGN

4.1 DATA FLOW DIAGRAM

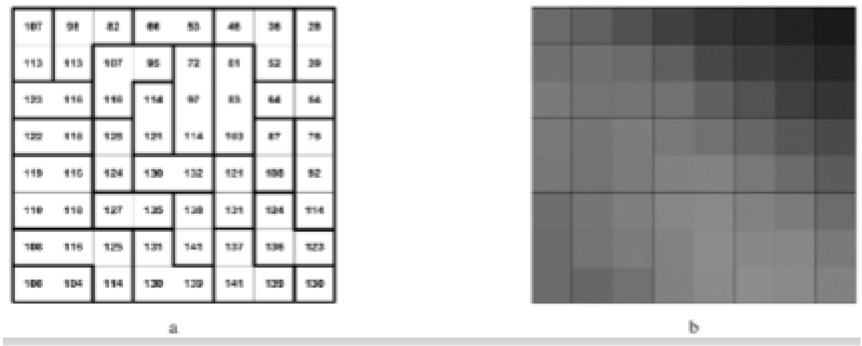
A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).



Training and testing data set: CIFAR dataset is a collection of images that are commonly used to train machine learning and computer vision algorithms. The CIFAR-10 dataset consistes of 60000 32X32 colour images in 10 classes, with 6000 images per class. There are 50000 training images based on which the model is trained. And the remaining 10000 images are used for testing.

Pre-processing techniques:

Normalization: In image processing, normalization is a process that changes the range of pixel intensity values. Normalization is an important step which ensures that each input parameter (pixel, in this case) has a similar data distribution. This makes convergence faster while training the network. Most probably the images are converted to gray scale as the pixels range will be 0-255 only.



One hot encoding:

One-hot is a group of bits among which the legal combinations of values are only those with a single high bit and all the others low.

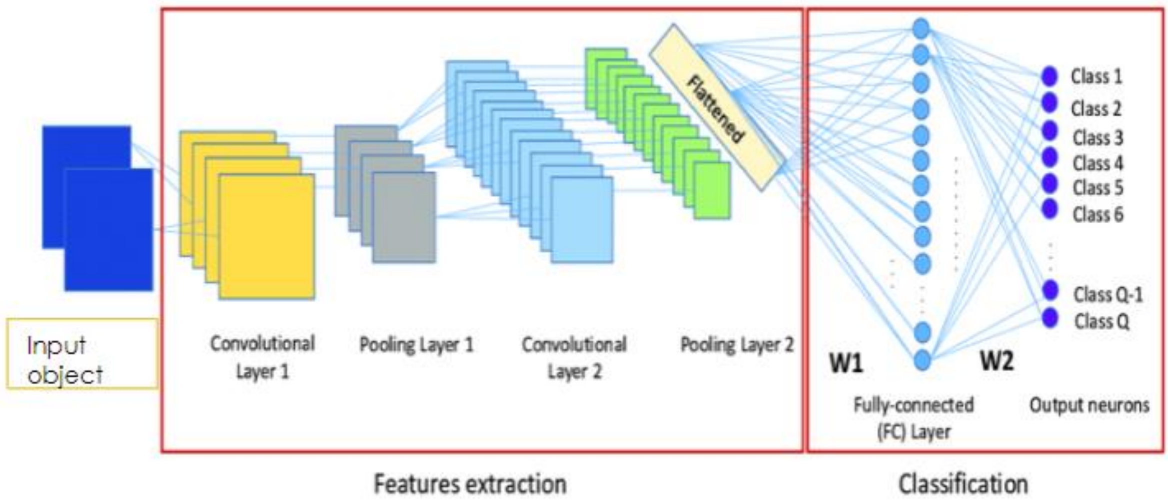
Color
Red
Red
Yellow
Green
Yellow

Red	Yellow	Green
1	0	0
1	0	0
0	1	0
0	0	1

Loss function: The error for the current state of the model must be estimated repeatedly. This requires the choice of an error function, conventionally called a loss function, that can be used to estimate the loss of the model so that the weights can be updated to reduce the loss on the next evaluation.

Epoch: An epoch refers to one cycle through the full training dataset. Usually, training a neural network takes more than a few epochs. In other words, if we feed a neural network the training data for more than one epoch in different patterns, we hope for a better generalization when given a new "unseen" input (test data). An epoch is often mixed up with an iteration. Iterations is the number of batches or steps through partitioned packets of the training data, needed to complete one epoch. As the number of epochs increases the accuracy of the model also increases with it.

Feature extraction: CNN is a neural network that extracts input image features and another neural network classifies the image features. The input image is used by the feature extraction network. The extracted feature signals are utilized by the neural network for classification. The neural network classification then works on the basis of the image features, the neural network for feature extraction includes convolution layer piles and sets of pooling layers.



CHAPTER 05

IMPLEMENTATION:

STEPS FOR IMAGE CLASSIFICATION :

1. Load the dataset from keras datasets module.

2. Plot some images from the dataset to visualize the dataset.

3. Import the required layers and modules to create our convolution neural net architecture.

4. Convert the pixel values of the dataset to float type and then normalize the dataset.

5. Now perform the one-hot encoding for target classes.

6. Create the sequential model and add the layers.

7. Configure the optimizer and compile the model.

8. View the model summary for better understanding of model architecture.

9. Train the model.

10. Calculate its accuracy on testing data.

The Library & Packages

Numpy:

NumPy is an acronym for "Numeric Python" or "Numerical Python". It is an open-source extension module for Python, which provides fast precompiled functions for mathematical and numerical routines. Furthermore, NumPy enriches the programming language Python with powerful data structures for efficient computation of multi-dimensional arrays and matrices.

Keras:

Keras contains numerous implementations of commonly used neural network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier.

TensorFlow:

TensorFlow is a Python library for fast numerical computing created and released by Google. It is a foundation library that can be used to create Deep Learning models directly or by using wrapper libraries that simplify the process built on top of TensorFlow.

Convolutional Layers:

The convolutional layers serve as feature extractors, and thus they learn the feature representations of their input images. The neurons in the convolutional layers are arranged into feature maps. Each neuron in a feature map has a receptive field, which is connected to a neighborhood of neurons in the previous layer via a set of trainable weights, sometimes referred to as a filter bank. Inputs are convolved with the learned weights in order to compute a new feature map, and the convolved results are sent through a nonlinear activation function. All neurons within a feature map have weights that are constrained to be equal; however, different feature maps within the same convolutional layer have different weights so that several features can be extracted at each location.

- **Pooling Layers:** The purpose of the pooling layers is to reduce the spatial resolution of the feature maps and thus achieve spatial invariance to input distortions and translations. Initially, it was common practice to use average pooling aggregation layers to propagate the average of all the input values, of a small neighborhood of an image to the next layer. However, in more recent models, , max pooling aggregation layers propagate the maximum value within a receptive field to the next layer.

- **Fully Connected Layers:** Several convolutional and pooling layers are usually stacked on top of each other to extract more abstract feature representations in moving through the network. The fully connected layers that follow these layers interpret these feature representations and perform the function of high-level reasoning. For classification problems, it is standard to use the softmax operator on top of a DCNN. While early success was enjoyed by using radial basis functions (RBFs), as the classifier on top of the convolutional towers found that replacing the softmax operator with a support vector machine (SVM) leads to improved classification accuracy.

➢ **Compiling Model:** Compile defines the loss function, the optimizer and the metrics. It has nothing to do with the weights and you can compile a model as many times as you want without causing any problem to pretrained weights. You need a compiled model to train (because training uses the loss function and the optimizer.

- **LOSS**: The error for the current state of the model must be estimated repeatedly. This requires the choice of an error function, conventionally called a loss function, that can be used to estimate the loss of the model so that the weights can be updated to reduce the loss on the next evaluation.

- **METRICS**: Metrics is used to evaluate the performance of our model. It is similar to loss function, but not used in training process. Keras provides quite a few metrics as a module, metrics and they are accuracy,binary accuracy, categorical accuracy and sparse categorical accuracy.

- **OPTIMIZER**: Optimization is an important process which optimize the input weights by comparing the prediction and the loss function. Keras provides quite a few optimizer as a module, optimizers. Adam is the best optimizers. If one wants to train the neural network in less time and more efficiently than Adam is the optimizer.

➢ **Training:** CNNs and ANN in general use learning algorithms to adjust their free parameters in order to attain the desired network output. The most common algorithm used for this purpose is backpropagation. Backpropagation computes the gradient of an objective function to determine how to adjust network parameters in order to minimize errors that affect performance. A commonly experienced problem with training CNNs, and in particular DCNNs, is overfitting, which is poor performance on a held-out test set after the network is trained on a small or even large training set. This affects the model's ability to generalize on unseen data and is a major challenge for DCNNs that can be assuaged by regularization. Here we use epochs during training the model and as shown in the below figure the accuracy of the model increases with the increase in the epochs value and the accuracy that we have achieved after having the epoch value 20 is 90.46%.

```
Epoch 1/20
391/391 [==============================] - 510s 1s/step - loss: 1.8541 - accuracy: 0.3054
Epoch 2/20
391/391 [==============================] - 507s 1s/step - loss: 1.1563 - accuracy: 0.5869
Epoch 3/20
391/391 [==============================] - 500s 1s/step - loss: 0.9290 - accuracy: 0.6742
Epoch 4/20
391/391 [==============================] - 493s 1s/step - loss: 0.8097 - accuracy: 0.7182
Epoch 5/20
391/391 [==============================] - 498s 1s/step - loss: 0.7213 - accuracy: 0.7488
Epoch 6/20
391/391 [==============================] - 491s 1s/step - loss: 0.6495 - accuracy: 0.7709
Epoch 7/20
391/391 [==============================] - 489s 1s/step - loss: 0.6097 - accuracy: 0.7886
Epoch 8/20
391/391 [==============================] - 486s 1s/step - loss: 0.5533 - accuracy: 0.8071
Epoch 9/20
391/391 [==============================] - 491s 1s/step - loss: 0.5013 - accuracy: 0.8235
Epoch 10/20
391/391 [==============================] - 487s 1s/step - loss: 0.4626 - accuracy: 0.8379
Epoch 11/20
391/391 [==============================] - 493s 1s/step - loss: 0.4455 - accuracy: 0.8413
Epoch 12/20
391/391 [==============================] - 491s 1s/step - loss: 0.4048 - accuracy: 0.8593
Epoch 13/20
391/391 [==============================] - 490s 1s/step - loss: 0.3858 - accuracy: 0.8629
Epoch 14/20
391/391 [==============================] - 496s 1s/step - loss: 0.3643 - accuracy: 0.8713
Epoch 15/20
391/391 [==============================] - 488s 1s/step - loss: 0.3324 - accuracy: 0.8819
Epoch 16/20
391/391 [==============================] - 488s 1s/step - loss: 0.3170 - accuracy: 0.8880
Epoch 17/20
391/391 [==============================] - 490s 1s/step - loss: 0.3061 - accuracy: 0.8901
Epoch 18/20
391/391 [==============================] - 492s 1s/step - loss: 0.2913 - accuracy: 0.8995
Epoch 19/20
391/391 [==============================] - 494s 1s/step - loss: 0.2777 - accuracy: 0.9017
Epoch 20/20
391/391 [==============================] - 487s 1s/step - loss: 0.2700 - accuracy: 0.9046
```

Final Predictions:

FINAL OBJECT DETECTION

```
fig, axs = plt.subplots(3, 3, figsize = (19,6))
fig.subplots_adjust(hspace = 1)
axs = axs.flatten()

for i, img in enumerate(batch):
    for key, value in class_labels.items():
        if class_result[i] == key:
            title = 'Prediction: {}\nActual: {}'.format(class_labels[key], class_labels[labels[i]])
            axs[i].set_title(title)
            axs[i].axes.get_xaxis().set_visible(False)
            axs[i].axes.get_yaxis().set_visible(False)

    # plot the image
    axs[i].imshow(img)

# show the plot
plt.show()
```

CHAPTER 06

CONCLUSION

By using this thesis and based on experimental results we are able to detect object more precisely and identify the objects individually. The best possible outcome in terms of accuracy by using Convolutional Neural Network. It is also shown that CNN doesn't need any translation of images into numerical data as CNN directly analyzes the images for training and testing purposes.

FUTURE ENHANCEMENTS

The object recognition system can be applied in the area of surveillance system, face recognition, fault detection, character recognition etc. The objective of this thesis is to develop an object recognition system to recognize the 2D and 3D objects in the image. The performance of the object recognition system depends on the features used and the classifier employed for recognition. This research work attempts to propose a novel feature extraction method for extracting global features and obtaining local features from the region of interest. Also, the research work attempts to hybrid the traditional classifiers to recognize the object. The object recognition system developed in this research was tested with the benchmark datasets like COIL100, Caltech 101, ETH80 and MNIST. The object recognition system is implemented in MATLAB 7.5.

It is important to mention the difficulties observed during the experimentation of the object recognition system due to several features present in the image. The research work suggests that the image is to be preprocessed and reduced to a size of 128 x 128. The proposed feature extraction method helps to select the important feature. To improve the efficiency of the classifier, the number of features should be less in number. Specifically, the contributions towards this research work are as follows,

- An object recognition system is developed, that recognizes the two dimensional and three-dimensional objects.
- The feature extracted is sufficient for recognizing the object and marking the location of the object. The proposed classifier is able to recognize the object in less computational cost.
- The proposed global feature extraction requires less time, compared to the traditional feature extraction method.
- The performance of the SVM-kNN is greater and promising when compared with the BPN and SVM.
- The performance of the One-against-One classifier is efficient.

Bibliography

This document contains provisions which, through reference in this text, constitute provisions of the present document.

1) Google Search Engine for various searching.

2) Introduction to machine learning, by Alex Smola and S.V.Nishwanathan.

3) Neural Networks and Deep Learning: By Michael Nielsen