Ummi Aishatu Ibrahim
940165
Coursework: object Recognition
Course: Big data and machine learning
11/12/2020

INTRODUCTION:

Object recognition is a technology in the field of computer vision, machine learning and image processing that detects or finds an object in an image or video. It has been one of the most researched problems in the computer vision and machine learning field. It is also one of the hardest problems to face. The task for this report is to achieve a suitable solution for this problem based on research.

When choosing the best algorithm for object recognition, one needs to ask certain questions such as what type of learning does it possess, what data is to be collected and printed, what is its feature representation, is the feature representation suitable for the project, what is its modelling, estimation and validation. Such questions should come first when dealing with computational machine learning problems.

There are a lot of compilated algorithms used for object recognition. Algorithms mostly used in supervised learning (CNN, SVM, etc), each algorithm has their own advantage why they are the most suitable for object recognition. Support Vector Machine is a learning that is associated with learning algorithms that analyse data used for classification and regression analysis. While, Convolutional Neural Network (CNN) algorithm is a deep neural network that is mostly used for analysing visual imagery.

However, I will be focusing an algorithm called Convolutional Neural Network (CNN) because it is considered to be one of the best algorithms to use for object recognition (Bhavsar, 2019) and it has made a huge contribution to object recognition. This is because it recognises each pattern, shape, colour and texture of an image or video which leads to an accurate image classification.

The results after this experiment will determine whether CNN is the best solution for the CIFAR dataset that was given for the project. We will determine the test accuracy, classification report and the confusion matrix of the dataset.

Method:

This section will cover the sub sections of how the project will succeed. It will explain how the features of the dataset will be extracted, processed, which classifier will be used and how the training and testing dataset will be trained and tested.

Feature extraction:

In CNN, there are three main functions needed when extracting features from a dataset, which are as follows; the convolution layer, the pooling layer and the classification layer, there are also some extra functions that can be added when extracting these features such as the dense layer and the flatten layer. Extraction occurs when there is a flow of data into a couple of convolutional and pooling layers. In this process the reduction of redundancy occurs as well. Then, all the features will be combined partially and sent to a fully connected layer where an estimate of the classification is provided.

Convolution layer:

This layer is the first stage when extracting features from a dataset, it is one of the most efficient ways of feature extraction and it makes sure it is always visible to computers. It is a simple application to a filter to an input that results in an activation. This layer compresses the input of the image by extracting the interest form which produces the feature maps. The feature maps are an output of a filter which is applied to a previous layer. This map is then responsed based on different feature detectors. Convolutional layers have different layers when extracting features. In the first layer, simple features (as edges) are filtered by neurons, these neurons gather as many information as possible from the images to create a bigger picture for clarity. Each convolution layer has various parameters. For example, the input size, the kernel size, the filters, depth of the map etc. In the program, I used the convolution 2-D layer because the images are 2-dimensional. The various parameters that I used are the filter, kernel size, input size and activation.

Activation function:

This function is present because it breaks any simple linear combination of the input and makes it possible for the neural network to become an approximator of a continuous function in the Euclidean space. It is also called the

non-linear activation function. It is used to deploy the output of neurons in an image. It is stated to be the best used and accepted function for activating convolutional outputs. It also prevents CNN from taking images as a linear function.

Pooling layer:

This function helps to identify images irrespective of any outdoor presence such as lightning, shade etc. It always interposes between a couple of convolutional layers. There are two types that are mostly used the Max pooling and the Average pooling. In my program I used the max pooling because it helps with overfitting by reducing data dimensionality and it focuses on the highest value of each feature map.

Flatten layer:

This feature transforms the entire pooled feature matrix after it has been pooled and compresses it into a single column. This process helps the neural network in terms of processing. It is then passed to the fully connected layer. This makes it easier to work with the images

Dense layer:

This function connects with the previous neurons by taking each neuron in a layer which receives an input from all the present neurons and densely connects them to the next layer.

Classification layer:

This layer is the final stage of the CNN which collects all the final features and returns them as a column vector. This layer makes sure that the CNN algorithm is fully connected, this is because it has an effective sample space projection which provides an obvious exhibition classification (Yu, 2018).

Feature processing:

Feature processing is the process of making your data more meaningful. In this project I implemented the dimensionality reduction as my feature processing. I used the train_test_split to divide the training dataset into images, labels and their validation to shuffle through the dataset. This helps to show the given class of the dataset to the model.

Classification:

For this project, the classification I will be using the CNN classification, considering I will be using deep neural networks for my project. This is because CNN is a deep learning algorithm, the classifier being used is the multiclass classifier. This classifier deals with neurons, then it multiplies it with its weight and run it on a non-linear activation function like ELU. Such classifiers take in input variables and passes it in the next layer continuous. This is the format that helps to solve complex problems in CNN because it is very effective for dimensionality reduction. It also fixes input and output variables and problems that deal with over-fitting and under-fitting.

Training and Testing in CNN:

Training:

The three main processes in training a CNN is by preparing the data, building the model and training the data. Training is done when the kernels are found in the convolutional layers and weights in the fully connected layers that minimizes the output between the ground truth labels and output predictions of the training set. When training a neural network, backpropagation algorithm is mostly used. This algorithm finds the minimum error value of a function in weight space by using a technique called the delta rule. This is considered to be the solution of the learning problem. The kernel and weight are then calculated by a loss function through the training dataset and are then updated according to the loss value of the algorithm.

The data is first loaded using NumPy and the data is changed to fit the CNN model using the transpose function. The Training set is then divided using the split function from model_slection in sklearn. The model for CNN is

then defined as the Sequential function where all the convolution, maxpooling flatten etc functions are accessible. They will then be defined according to the specification of CNN. The summary will then be printed before the compilation of the model is then defined. In the compilation, optimizer, loss and metrics are defined to fit the model for accuracy. Then model will be fitted with the training set using the testing set as its validation data.

Testing:

The testing process of CNN mostly compiles of the accuracy for the success for the object recognition. The testing for CNN model is done at the evaluation stage. The evaluation stage takes in the testing dataset and prints out the final test loss and accuracy for the project. The benchmark accuracy given for the project is about 44.68%, however it is not a targeted accuracy. Once the testing is complete the project has succeeded.

Results:

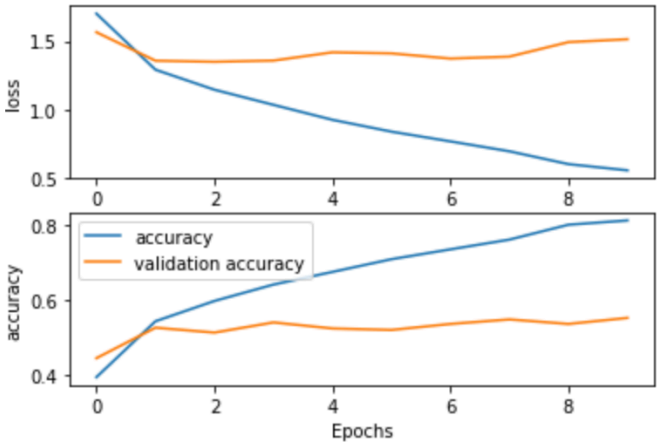
In this section I will be discussing the experimental results for the test program that was assigned to us. As said, I used the CNN algorithm to determine the accuracy of object recognition in the CIFAR 10 dataset. From the program after training and evaluating the testing datasets, I was able to come up with an accuracy of 55.199% accuracy. It is not an encouraging percentage; however, we are dealing with a large dataset and losses can arise when training the data which leads to the reduction of test accuracy.

After trying to fit the model using the training set and testing set as the validation data, I was able to come up with the test accuracy for the epoch of the 10 categories of the CIFAR 10 dataset.

```
Epoch 1/10
157/157 [==============================] - 4s 29ms/step - loss: 1.7022 - accuracy: 0.3945 - val_loss: 1.5658 - val_accuracy: 0.4450
Epoch 2/10
157/157 [==============================] - 4s 28ms/step - loss: 1.2908 - accuracy: 0.5431 - val_loss: 1.3566 - val_accuracy: 0.5260
Epoch 3/10
157/157 [==============================] - 4s 28ms/step - loss: 1.1438 - accuracy: 0.5971 - val_loss: 1.3491 - val_accuracy: 0.5130
Epoch 4/10
157/157 [==============================] - 4s 27ms/step - loss: 1.0327 - accuracy: 0.6407 - val_loss: 1.3576 - val_accuracy: 0.5400
Epoch 5/10
157/157 [==============================] - 4s 28ms/step - loss: 0.9228 - accuracy: 0.6744 - val_loss: 1.4184 - val_accuracy: 0.5240
Epoch 6/10
157/157 [==============================] - 4s 27ms/step - loss: 0.8359 - accuracy: 0.7082 - val_loss: 1.4105 - val_accuracy: 0.5200
Epoch 7/10
157/157 [==============================] - 4s 27ms/step - loss: 0.7644 - accuracy: 0.7344 - val_loss: 1.3723 - val_accuracy: 0.5360
Epoch 8/10
157/157 [==============================] - 5s 30ms/step - loss: 0.6916 - accuracy: 0.7603 - val_loss: 1.3869 - val_accuracy: 0.5480
Epoch 9/10
157/157 [==============================] - 4s 28ms/step - loss: 0.5974 - accuracy: 0.7999 - val_loss: 1.4933 - val_accuracy: 0.5360
Epoch 10/10
157/157 [==============================] - 4s 27ms/step - loss: 0.5519 - accuracy: 0.8112 - val_loss: 1.5142 - val_accuracy: 0.5520
```

This diagram is the result given after training the CIFAR 10 dataset using an epoch of 10.

After producing the epoch for the dataset, I plotted the graph for the epoch against loss and accuracy to display the representation of the results.



As shown in the diagram the accuracy is going well as it increases. As the accuracy increases the losses reduces. This makes sure that the object recognition of each image is quite good making CNN algorithm a perfect algorithm when dealing with object recognition.

The confusion matrix and classification report for the report is as follows:

```
              precision    recall  f1-score   support

    airplanes       1.00      0.15      0.26      1000
   automobile       0.00      0.00      0.00         0
         bird       0.00      0.00      0.00         0
          cat       0.00      0.00      0.00         0
         deer       0.00      0.00      0.00         0
          dog       0.00      0.00      0.00         0
         frog       0.00      0.00      0.00         0
        horse       0.00      0.00      0.00         0
         ship       0.00      0.00      0.00         0
        truck       0.00      0.00      0.00         0

     accuracy                           0.15      1000
    macro avg       0.10      0.02      0.03      1000
 weighted avg       1.00      0.15      0.26      1000

[[151  87  93  75  78 108  83 141  91  93]
 [  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]]
```

As seen above the is showing the performance and predictions of CNN algorithm on object recognition

Conclusion:

In conclusion, object recognition is one of the hardest problems being faced in the computer vision and machine learning field. However, everyone is doing their best to overcome it and produce the best algorithm that is suitable and understandable. The task given was produce and explain the best suitable algorithm for object recognition and I chose Convolutional Neural Network (CNN). This is because after countless research it is said to be the best out of the remaining algorithms.

CNN algorithm is quite understanding and difficult at the same time. The method is follows on feature extraction is very compilated when not understood. It extracts its feature from a dataset by using layers such as the convolution layer, the maxpooling layer, Dense etc. All these layers help to train the dataset before checking for its testing accuracy. It also uses the concept of PCA which is the Train_test_split function to shuffle the data to easier train the model and reduce dimensional redundancy. It is also classified under the multiclass classifier in the deep learning network.

The training and testing process of CNN is very straight forward, it is trained using a convolutional layer and weights in a fully connected layer to minimize its output between its ground truth and predictions. It also tested using its accuracy and loss to determine the success of the object recognition process.

The shortcomings for this project are the concepts to fully understand the CNN algorithm. CNN is a deep learning algorithm which makes it harder to work with, however its easier for object recognition. Another issue is the concept to use the layers for the program, during research I found out that the more the layers, the more the test accuracy. However, this does apply to the CIFAR 10 datasets. I tried to implement this concept, but the test accuracy kept going lower and lower.

CNN algorithm could use more research and work to fully understand its concept. Object recognition is just at the process of getting better because the world is slowly moving into technology and more people are starting to get interested in its field. My solution to this problem is to keep comparing different algorithms and keep trying to imply it in different object datasets, that way its test accuracy might go higher and there will be an improvement or possibly new algorithms that will focus more on object recognition.

Bibliography

Bhavsar, P. (2019, January 18). *Data Driven Investor*. Retrieved from Object Detection with Convolutional Neural Networks: https://medium.com/datadriveninvestor/object-detection-with-convolutional-neural-networks-dde190eb7180

Cascianelli, s., Cerezo, R. B., Bianconi, F., Fraviolini, M. L., Belala, M., Palumbo, B., & Kather, J. N. (2017). Dimensionality Reduction Strategies for CNN-Based Classification of Histopathological Images. *Intelligent Interactive Multimedia Systems and Services 2017*, 21-30.

Rampurawala, M. (2018, Febuary 8). *Classification with TensorFlow and Dense Neural Networks*. Retrieved from Heartbeart: https://heartbeat.fritz.ai/classification-with-tensorflow-and-dense-neural-networks-8299327a818a

Yamashita, R., Nishio, M., Do, G. R., & Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. *Springer Open*, 611-629.

Yu, H. L. (2018). Feature Extraction and Image Recognition with Convolutional Neural Networks. *Jornal of Physics: Conference series*, 1-8.