Artificial Intelligence Image Recognition

Aman Dureja
Dronacharya College Of Engineering, Khentawas, Gurugram, Haryana, India

***Abstract-* In this research paper, we will look at image recognition notion and its use in Face Detection, Google lens, etc. Image recognition is the procedure of recognizing and detecting an object in a digital image. Also, we will look at a very small image recognition model and see how difficult it is to make our very own image-recognition model. The image-recognition technique is used in various sectors like security surveillance, toll booth monitoring and self-driving cars but in my eyes its most productive use is in language detection and translation to make easier communication between people. Image recognition is a part of computer vision and a process to identify and detect an object or attribute in a digital video or image. Computers can use machine vision technologies in combination with a camera and artificial intelligence software to achieve image recognition. Computer vision is a broader term which includes methods of gathering, processing and analyzing data from the real world. It becomes very difficult for people who travel in different countries to communicate with locals, this is a huge advancement if we could just take a photo and the machine recognizes and translates the language. The extraction of image features is one of the fundamental tasks in image recognition.**

***Index terms-* Image recognition, Machine learning, Modules, Keras libraries, Deeper neural networks**

I.INTRODUCTION

Image recognition is the technique in which we simply take a photo of something and the machine realises what the object is, this approach is used in many places like Google lens, translate visual content for blind users etc. We need this method more in our world as there are many languages, where we can use image recognition to make an application which can identify scripts of various languages and can convert it into other languages, it is also in practice in form of Google translate, and some other applications, but there are many language scripts which are yet to be learnt by the machine. This process could take a lot of time as the AI will learn each alphabet of a language then meaning and lots of coding is also required. Machine learning has been successfully used to achieve state-of-the-art performance in a variety of applications such as web search, spam detection, caption generation, and speech and image recognition. Generating tests is also a challenge. Unless the number of potential tests is huge or the material being tested is highly dynamic, one runs the risk of an adversary generating all possible tests and using a hash function to look up the answer in a precomputed database. We must find a test with a new property, the test must be easy to generate but intractable to pass without special knowledge available to humans and not computers.

II. LITERATURE SURVEY

[1]It becomes very difficult for deeper neural networks to be trained, so to reduce the difficulty a residual learning framework is presented [2] there are many problems with rotation-, scale-, and translation-invariant features. To fix this a set of rotation-invariant features are introduced [3] one of the fundamental tasks in image recognition is extraction of image, there are other features namely visual features, statistical features of pixel, transform coefficient features. [4]An image recognition system using end to end deep learning is presented with a highly optimized parallel algorithm.

III. SOME IMAGE RECOGNITION TECHNIQUES

[1] There is an Image Net dataset, organized according to the WordNet hierarchy where residual nets are evaluated with a depth of up to 152 layers i.e., 8x deeper than VGG nets but still it has lower complexity. Each WordNet is also known as synset or synonym set. In ImageNet, our aim is to provide on average 1000 images to interpret each synset. All of these residual nets can archive 3.57% error on the image net test. The image is resized with its shorter

side randomly sampled for scale augmentation and per-pixel mean subtracted.

[2] For translation uniformity there should be regular movements of each image. These movements should be utilized. Also, translational invariances could be achieved by transforming the images into new images having first order moments are equal to zero.

[3] Data parallelism can work for smaller models but in case of much bigger models, they cannot fit into memory of a single GPU. Here model parallelism can help, this method is still in use at convolutional layers but fully connected layers are instead partitioned and distributed to multiple GPUs.

[4] Image editing and Image restoration are also important techniques used by image recognition. In image editing the digital images are altered through graphic software tools whereas in image restoration we can estimate the no of clean image formed from corrupt image taken in order to get back the information lost.

IV. A SIMPLE IMAGE RECOGNITION MODEL USING PYTHON

Image recognition model can be made by using the following five steps in an appropriate manner:

A. Import modules, classes, and functions

In the following code we will use a keras library for handling the neural network and scikit-learn to get and prepare data.

B. Load data

In this step, a machine learns to recognize the hand written digits. The function load_digits() from sklearn. datasets provide 1797 observations. Each observation has 64 features which represents the pixels of 1797 pictures 8px high and 8px wide

C. Transform and split data

In the third step, we have to convert the output in binary form and then we will split the entire dataset into training and test sets. Then standardize the outputs.

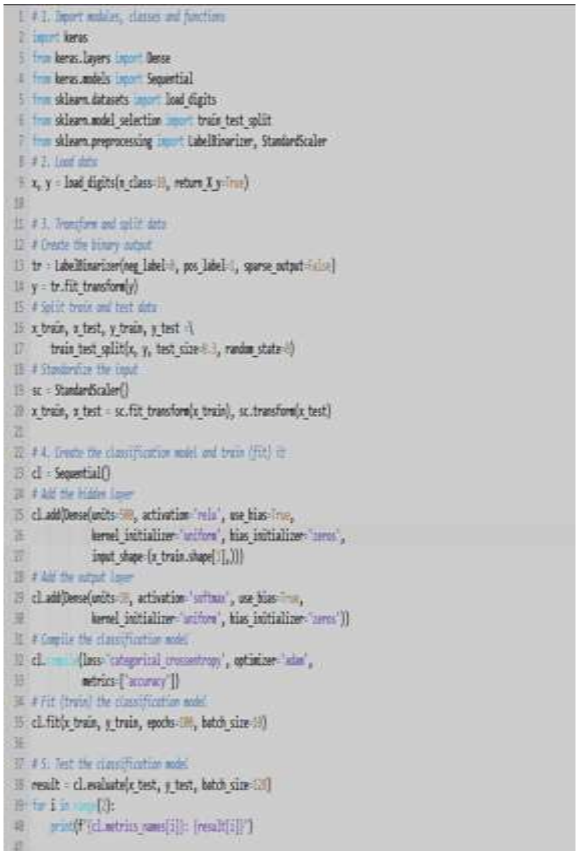
D. Creation of classification model and its training(fit)

The models which are simple will have one input layer (which is not explicitly added), one hidden layer and one output layer. We should use a training set in order to train our neural network.

E. Testing of classification model

Now we will test the performance of our network, using the test set.

V. CODE



```
# 1. Import modules, classes and functions
import keras
from keras.layers import Dense
from keras.models import Sequential
from sklearn.datasets import load_digits
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelBinarizer, StandardScaler
# 2. Load data
x, y = load_digits(n_class=10, return_X_y=True)

# 3. Transform and split data
# Create the binary output
tr = LabelBinarizer(neg_label=0, pos_label=1, sparse_output=False)
y = tr.fit_transform(y)
# Split train and test data
x_train, x_test, y_train, y_test =\
    train_test_split(x, y, test_size=0.3, random_state=0)
# Standardize the input
sc = StandardScaler()
x_train, x_test = sc.fit_transform(x_train), sc.transform(x_test)

# 4. Create the classification model and train (fit) it
cl = Sequential()
# Add the hidden layer
cl.add(Dense(units=64, activation='relu', use_bias=True,
             kernel_initializer='uniform', bias_initializer='zeros',
             input_shape=(x_train.shape[1],)))
# Add the output layer
cl.add(Dense(units=10, activation='softmax', use_bias=True,
             kernel_initializer='uniform', bias_initializer='zeros'))
# Compile the classification model
cl.compile(loss='categorical_crossentropy', optimizer='adam',
           metrics=['accuracy'])
# Fit (train) the classification model
cl.fit(x_train, y_train, epochs=100, batch_size=10)

# 5. Test the classification model
result = cl.evaluate(x_test, y_test, batch_size=128)
for i in range(2):
    print(f'{cl.metrics_names[i]}: {result[i]}')
```

Fig. 1 A code written for image recognition using keras libraries in Python.

VI. RESULT

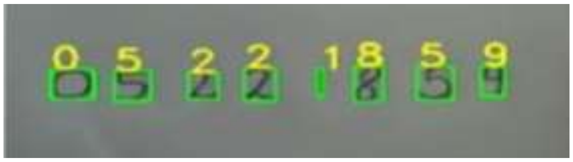


Fig. 2 Testing of the code written in Python

VII. APPLICATIONS

a. Improvement in Iris Recognition

b. Searches can now be done virtually for enhanced product discoverability
c. Facial and Image Recognition for Social platforms as well as for security purposes
d. A vision to machines and self-driving cars
e. Augmented Reality Gaming could improve to next level
f. Teaching machines for some work
g. Security Surveillance
h. Retail and Advertisement Industry might benefit from Image Recognition

VIII. CONCLUSIONS

At last, I would like to conclude that the following model is a very minor example of image recognition. To build an application which can recognize different languages and different vernacular is a clumsy task, so to build that challenging model respective team members are required, the following code is written with the help of python and keras libraries. We can make other models also with the help of Artifical Intelligence.

REFERENCES

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun; The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778
[2] Alireza Khotanzad, Yaw Hua Hong IEEE Transactions on pattern analysis and machine intelligence 12 (5), 489-497, 1990
[3] Zi-Quan Hong Pattern recognition 24 (3), 211-219, 1991
[4] Ren Wu, Shengen Yan, Yi Shan, Qingqing Dang, Gang SunarXiv preprint arXiv:1501.02876 7 (8), 2015