29/02/2020 **CIA – 3** (Component 2) 1740256

Jeevan Koshy

--------------------------------------------------------------------------------------------------------------------------------------

**IMAGE PROCESSING USING NEURAL NETWORKS**

--------------------------------------------------------------------------------------------------------------------------------------

**Concept**

Off late, it is the trend now with all the buzz words in the tech industry which consist of Machine Learning, Data Science, Cyber-Security. Image processing is one of them and there are so many use cases and applications where this domain is used.

The theory behind this concept is that a digital image is processed by means of a digital computer. Image processing would result in benefits such as extracting useful information from the enhanced and processed image. We would get these results only after analysing and manipulation of the image.

An image is a two – dimensional array that is arranged in the form of rows and columns. These consist of finite number of elements that have different values at corresponding locations. The term pixel is used to denote these elements. There are different types of images that are used for processing. They consist of –

* Binary Image – Consists of 2-pixel elements ~ 0 refers to black and 1 refers to white.
* Black & white Image – Only black and white colours.
* 8-bit colour format – 256 different shades of colours.
* 16-bit colour format – 65,536 colours.

This field implements AI technology to automatically identify different objects, images, places or things. Operations such as labelling images with descriptive tags, searching for content in images, guiding robots, autonomous vehicles and the like. The ability to recognise and image is natural for human beings but difficult for a computer system.

Just like the human eye sees an image and processes the image in the brain, the computer sees the image as a set of vectors. Computers can logically analyse the image by simplifying the images, extracting important information and organising data through feature extraction and classification. The vision of these computers then use classification or any other algorithm to convey a decision of parts of the image or even the whole image.

**Algorithm**

*The following steps describe the procedure of processing our images –*

1. Pre – processing

Some Sample pre – processing techniques are that we can crop parts of the image, flip the image horizontally and also adjust hue, contrast and saturation. There are different techniques of pre – processing which can be applied. The objective of pre – processing is to improve, rebuild or restore images.

There are a few parameters which have to be considered for data preparation –

1. Size of the image
2. Number of images
3. Number of channels
4. Aspect ratio
5. Image scaling
6. Mean, standard deviation of input data
7. Normalising input images
8. Dimensionality reduction
9. Data augmentation
10. Splitting the dataset

We can split the dataset into 2 different subsets such as training and testing where the training dataset has majority of the images so that the algorithm can learn what kind of images are being fed for it to predict when the later images come. Data reduction or feature extraction are other terms in which the image can be compressed to extract geometric characteristics.

1. Building a convolutional neural network

The beauty of image processing is that it can recognise images later on if we supply it with enough of training data. Rather than feeding the entire image as an array of numbers, the image breaks up into different tiles and the algorithm tries to predict what each tile is. The computer then tries to predict the picture based on what each tile is.

A convolutional neural network uses a 3 – dimensional structure where each set of neurons analyses a particular region of feature of the image. For example, in a picture of a cat, different groups of neurons analyse various parts of the cat such as the head, body, legs etc.

**Significance & Relevance**

This exciting field gives improved pictorial information for human clarification and processing of raw data.

Some applications of where it is used today are –

* Image enhancement
* Image restoration
* Image compression
* Character recognition
* Signature verification
* Biometrics
* Fingerprint verification
* Object recognition
* Traffic monitoring
* Face detection

**Code**

*import matplotlib.pyplot as plt*

*from PIL import Image*

*im=Image.open("test5.jpg")*

*im=im.convert('RGB')#Convert image to RBG color space*

*pix=im.load()*

*width,height=im.size#image dimensions*

*#Function to convert RGB to HSV*

*#HSV is more versatile when compared to RGB when working with image processing*

*def rgb\_to\_hsv(r, g, b):*

*maxc = max(r, g, b)*

*minc = min(r, g, b)*

*v = maxc*

*if minc == maxc:*

*return 0.0, 0.0, v*

*s = (maxc-minc) / maxc*

*rc = (maxc-r) / (maxc-minc)*

*gc = (maxc-g) / (maxc-minc)*

*bc = (maxc-b) / (maxc-minc)*

*if r == maxc:*

*h = bc-gc*

*elif g == maxc:*

*h = 2.0+rc-bc*

*else:*

*h = 4.0+gc-rc*

*h = (h/6.0) % 1.0*

*return h, s, v*

*GREEN\_RANGE\_MIN\_HSV = (80, 100, 100)#min range of green values*

*GREEN\_RANGE\_MAX\_HSV = (185, 255, 255)#max range of green values*

*for x in range(width):#traverse each pixel horizontally*

*for y in range(height):#traverse each pixel vertically*

*r, g, b = pix[x, y]*

*h\_ratio, s\_ratio, v\_ratio = rgb\_to\_hsv(r / 255.0, g / 255.0, b / 255.0)#function call to convert RGB values*

*h, s, v = (h\_ratio \* 360, s\_ratio \* 255, v\_ratio \* 255)*

*min\_h, min\_s, min\_v = GREEN\_RANGE\_MIN\_HSV#assigning min h,s,v values*

*max\_h, max\_s, max\_v = GREEN\_RANGE\_MAX\_HSV#assigning max h,s,v values*

*if min\_h <= h <= max\_h and min\_s <= s <= max\_s and min\_v <= v <= max\_v:#check if pixel color is within the range*

*pix[x, y] = (255, 255, 255,1)#set pixel value to white*

*plt.imshow(im)*

*plt.show()*

--------------------------------------------------------------------------------------------------------------------------------------