Image Recognition

Discrete Mathematics Project

Name: Kathan Sanghavi
ID: 201901053

June 5, 2020

1. Motivation

Image Recognition technique can help us in many ways. Image Recognition technique can be used in handwritten character recognition, we can scan the handwritten notes to save them digitally in text format. In the same way, technique can also be used to scan the printed book or printed paper and save notes digitally from them in text format. Image Recognition technique can be used as cheque reading system. Image Recognition can be used in medical applications and also to design smart glasses for visually impaired people. Image Recognition technique can also be used for facial recognition, which can be used on social media sites for tagging people. Image Recognition and Facial Recognition technique can also be used in security systems. For image recognition, our motive is to design technique which should act in similar way the vision processing works in our brain. Therefore we should design graph network inspired from working of neurons in our brain. Images are stored digitally as a matrix of pixel values therefore with use of matrix operations, we can extract patterns from images and we can use graph network for training and classification. Therefore problem of image recognition can be solved by using Matrix Theory and Graph Theory, Image Recognition technique can be designed using knowledge of Discrete Mathematics. Structure used for solving problem of image recognition is called Convolutional Neural Network. In the section 2 and 3, I have described technique of Image Recognition using Convolutional Neural Network. I have described other uses of Image Recognition, applications of Convolutional Neural Network and how the structure similar to the image recognition can also be used in natural language processing with the use of regular expressions in section 4 and 5. I have also written the code for image recognition and natural language processing in python, I have uploaded these codes on website.

2. Formulating Mathematics

Convolutional Neural Network can be trained to classify image. Process of classification of image can be broadly divided into two processes, feature extraction and classification. Images are stored as 2d array of pixels values. For grayscale images, the pixel value is typically an 8-bit data value (with a range of 0 to 255) or a 16-bit data value (with a range of 0 to 65535). For color images, there are 8-bit, 16-bit, 24- bit, and 30-bit colors. The 24-bit colors are known as true colors and consist of three 8-bit pixels, one each for red, green, and blue intensity. Therefore, feature extraction from image can be done by matrix operations. Once a feature has been detected by matrix operations on input image, its exact location becomes less important. Only its approximate position relative to other features is relevant. Precise location of features is irrelevant and taking precise location of feature into consideration can corrupt process as exact position of feature can vary for different images of same object. For example, images of animal cheetah can contain different background locations, images can be tilted, shifted and scaled. One important feature to classify image as cheetah's image is the black 'tear marks' running from the corners of the eyes down the side of the nose (Figure: 2). Identification of colour pattern is also a part of feature extraction. For example, detection of black and white stripes is a important feature extraction from image to classify image as zebra's image. In hand written character recognition, if by feature extraction we came to know that image contains the endpoint of a roughly horizontal segment in upper left area, a corner in the upper right area and the endpoint of a roughly vertical segment in the lower portion of the image, we can tell that the character should be number 7. From above examples, it is apparent that to do image classification, we need feature extractor module which takes image as input and gives feature vector as output. We need to apply series of matrix operations on input image to extract feature vector from it. Therefore feature extraction module should have layered structure, where each layer does defined matrix operation. From feature vector, classification of image can be done using trainable classifier module.

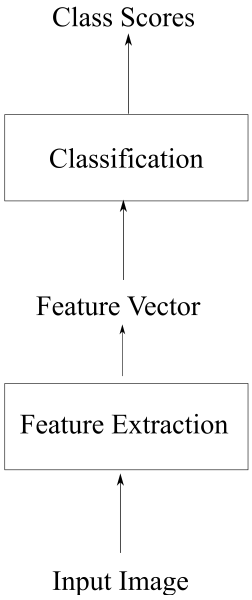

Figure 1: Image Recognition Process

There are many classification methods available such as K-Nearest Neighbors(K-NN), Support Vector Machine(SVM), Naive Bayes, Random Forest Classification and use of Artificial Neural Network. Weighted graph approach(Figure: 3) and training based on backpropagation used in Artificial Neural Network has advantages over other methods. Convolutional Neural Network can learn efficiently

Figure 2: Images of Cheetah
Source:Wikipedia

form training sets without having problem of over parameterized.

3. Solving Mathematics

From requirements stated earlier, it is clear that we should have multi-module system for image recognition. Feature extractor broadly have corresponding modules for following operations: (1) Convolution (2) Rectified Linear Unit (3) Pooling (4) Flattening.

First, we normalize the input image. Normalization of image is a process to change the range of pixel intensity values. The linear normalization of grayscale and RGB images is performed according to formula,

$$OutputChannel = 255 \times \frac{InputChannel - min}{max - min} \tag{1}$$

Equation 1 is for 8-bit data value channel. For x bit data channel, 255 should be replaced by $2^x - 1$. In the equation 1, max denotes maximum pixel value occurred in Input Channel and min denotes minimum pixel value occurred in Input Channel. As grayscale images contain only one channel, equation 1 is only applied to that one channel. For RGB images, images are normalized using equation 1 for each of three - Red, Green, Blue channels (Figure: 4). Input images should be scaled down or up,

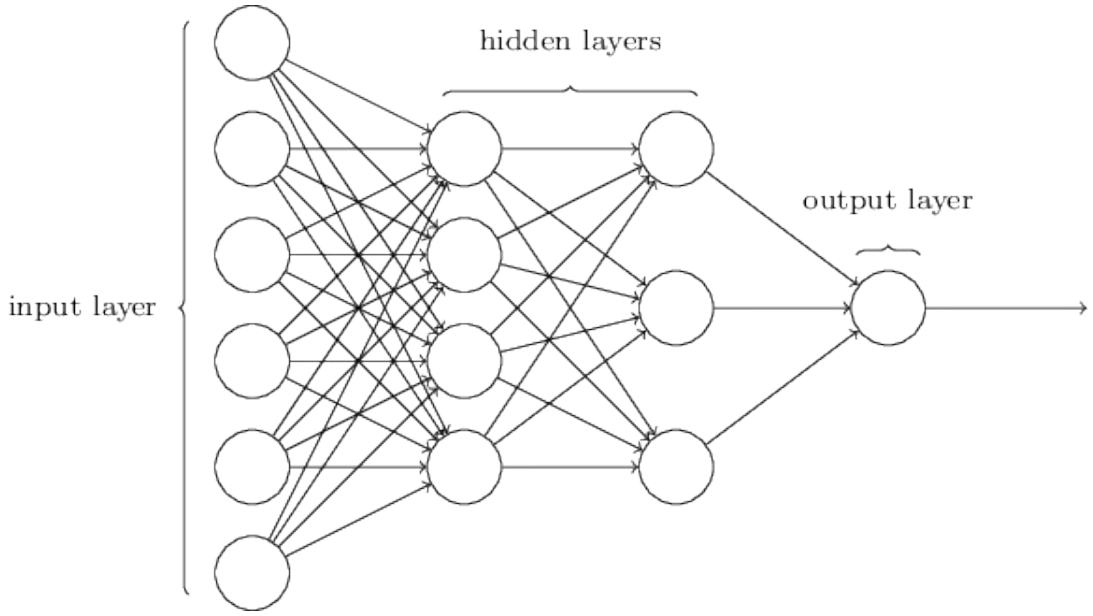

Figure 3: Weighted Graph approach for Classification
In the figure, weights are not shown but edges have weights associated with them. Figure is only for the representation of structure, number nodes can vary according to application.

if image is too large or too small. There are techniques such as upsampling through interpolation, padding the image using zeros for too small images and downsample, appropriately crop down to input size for too large images. Normalized and appropriately scaled image is given to convolution module as input.

A. Convolution

Feature Detector is a matrix used to detect whether certain feature is present in image or not. Feature Detector can be 3×3 or 5×5 or 7×7 matrix. Input image is convolved with feature detector. Convolution operation is performed according to formula,

$$C_{i,j} = \sum_{x=1+s(i-1)}^{n+s(i-1)} \sum_{y=1+s(i-1)}^{n+s(i-1)} A_{x,y} \times B_{x-s(i-1),y-s(i-1)} \tag{2}$$

In the equation 2, $1 \leq i \leq \lceil \frac{l-n}{s} \rceil + 1$, $1 \leq j \leq \lceil \frac{w-n}{s} \rceil + 1$ (indexing starts from 1), l is the number of rows in input image, w is the number of columns in input image, matrix A is input image, matrix B is feature detector and matrix C is output matrix, which is called feature map. Stride is denoted by s. Value of stride can be 1, 2, 3 or other integer value based on the application, as higher value of stride reduces the size of feature map which is good for too big images. Number of rows in feature detector is denoted by n. (Feature detector is taken as a square matrix). If $x > l$ or $y > w$, $A_{x,y} = 0$. Convolution operation can be formulated in algorithm as follows.

1. Set the location of placing result in output matrix as 1st row,1st column.

Figure 4: Normalization

2. Take the feature detector put it on image such that it covers top left corner - nine pixels (for 3×3 feature detector) in top left corner.

3. Repeat the following process till the end of row is reached.

   (a) Repeat the following process till the end of columns is reached.

   - Multiply each value of feature detector by corresponding (covered) pixel value in input image.
   - Sum up the result of this element wise multiplication.
   - Put the resultant sum at location of placing result.
   - Move feature detector stride columns right side.
   - Set the location of placing result as same row, next column.

   (b) Move the feature detector to stride rows down and to the left most columns.

   (c) Set the location of placing result as next row, 1^{st} column.

4. Output matrix is called Feature Map.

Convolution operation between two matrices is denoted by $\otimes$. Feature Detector is also called Filter or Kernel. Feature Map is also called Convolved Feature or Activation Map (Figure: 5).

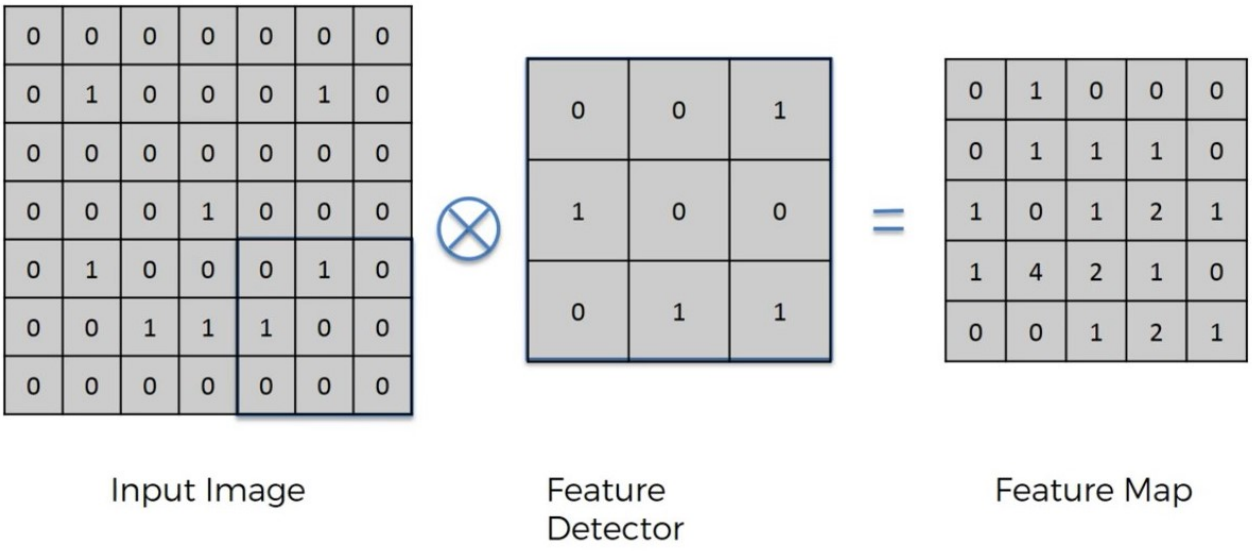

Figure 5: Convolution Operation
Input image and feature detector can have any integer pixel values. Here as an example, pixel values are 0 and 1, value of stride is 1, feature detector is taken as 3×3 matrix.

Principal Component Analysis - PCA can be used to reduce dimensionality. PCA is based on eigen vector and eigen values. Here, I am not using the Principal Component Analysis, but it can also be used according to application. Convolution module creates multiple feature maps by using many different feature detectors. Network decides which features are important through training. Convolution layer extracts important features from image and also reduces size, as output matrix contains important features from input image and also the size of output matrix is smaller than input image.

Here, as a example I am demonstrating Edge Detect feature detector (Figure: 6). Edge Detect feature detector is 3×3 matrix.

$$EdgeDetect = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

B. Rectified Linear Unit

Activation functions are used to increase non-linearity, as original images can have non-linearity but the use of convolution operation creates risk that we have created something linear. There are many activation functions available such as Sigmoid, TanH, Binary step, ArcTan, Rectified Linear Unit - ReLU. For our task ReLU is the most suitable one. Rectifier function is defined as $f(x) = max(0, x)$. There are also several other variants of ReLU. Feature maps are passed through ReLU layer.

C. Pooling

Figure 6: Applying Edge Detect feature detector to an image

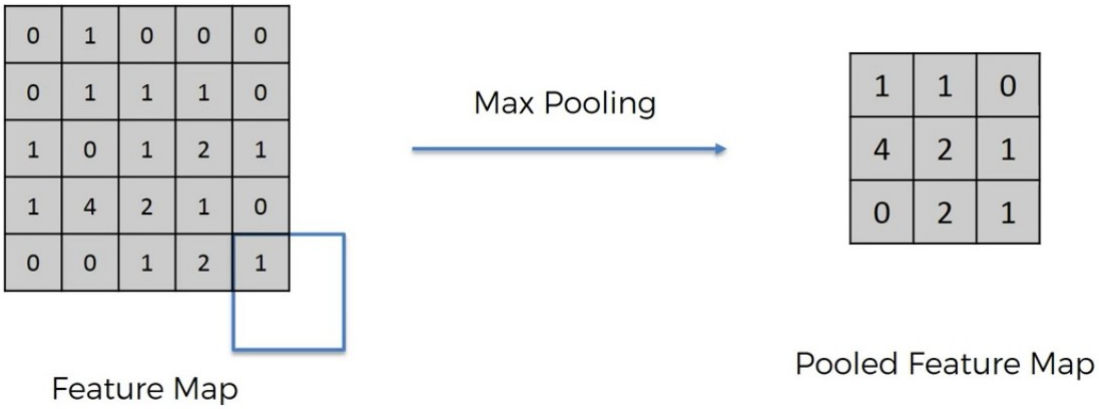

Figure 7: Max Pooling

Feature map records precise position of features in input. Therefore small movements in the position of the feature in the input image can result in a different feature map. This can also happen due to rotation, shifting, tilting and other minor changes in the input image. Therefore we must take only the relative location of features into consideration. With the use of Pooling module, we achieve reduction in size and we are able to take out only the relative locations of features. Pooling can be maximum pooling or mean pooling. In pooling operation, we take a window of size 2×2, (different size can also be taken according to application) and put the window on top left corner such that it covers 4 pixels. Take the maximum value from covered pixels as result and fill the output matrix in same way as described in algorithm of convolution operation. If some part of the window is out of the input matrix, then take the maximum value from cells covered by window. (which is also shown in figure 7). In mean pooling, output matrix is filled with mean of covered pixels by window. Pooling module reduces the number of parameters that finally goes in graph network, therefore also prevents overfitting.

D. Flattening

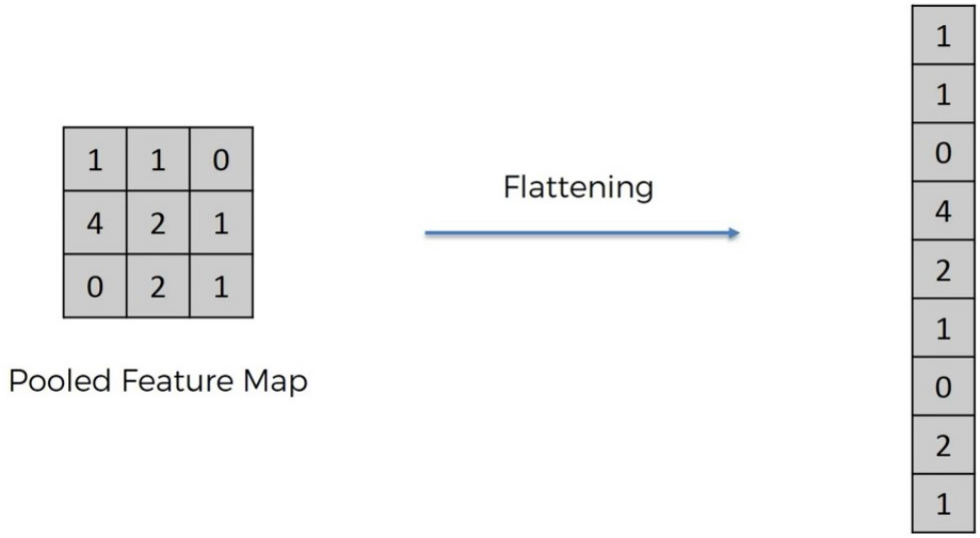

Figure 8: Flattening

In the flattening part, pooled feature map is flattened into one column vector. This column vector is given as input to graph network (Figure: 8).

Weighted Graph approach for Classification

I am using directed weighted graph network for classifier module. Node is the basic element for the classification using graph network. Function of node is described as follows. Node takes many inputs, which come from other nodes via edges coming in to the node. As we are using weighted graph, edges have certain weights associated with them. Graph network are able to learn to classify images by adjusting weights associated with edges. Output of node A is given according to formula, (also shown in figure 9)

$$Output = \phi\Big(\sum_{i=1}^{m} w_i x_i\Big) \tag{3}$$

In equation 3, m is the number of edges coming in to the node A, w_i denotes weight of edges which are coming in to the node A, x_i is the value of nodes from which edges are coming out and going in to the node A. Output of a node can also be considered as value of that node. ϕ denotes activation function. As stated earlier, activation function can be of many types such as rectifier, sigmoid, threshold, tanh. Activation function for a node is chosen according to which layer node is located in. Activation function also depends on output type of that node, if we want node to give binary outcome then sigmoid or threshold activation function is more preferable than others. Layer is a 1d array of nodes, nodes from same layer can't have edge connecting to each other. A trainable Graph Network is made using this layers. In general, Graph Network can be divided into three type of layers, (1) Input layer (2) Hidden layers (3) Output layer (Figure: 10). Input layer contains input parameters as nodes.

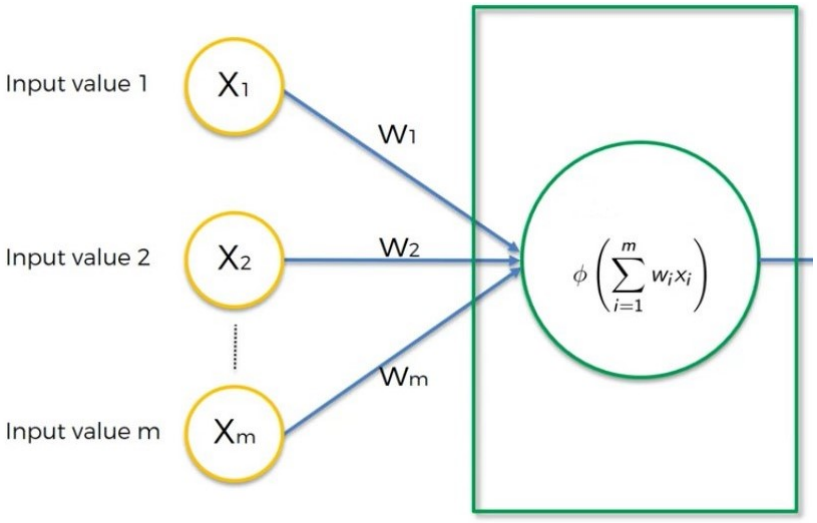

Figure 9: Nodes
Source:SlideShare

Layers located between input layer and output layer are called hidden layers. In case of Image Recognition every node has a connection using weighted edge to each node in next layer. Therefore in case of image recognition, hidden layers can also be called as fully connected layers. For graph network of Image Recognition, Hidden layers have rectifier activation function for nodes in them. Output layer have sigmoid activation function for nodes in it.

Training of graph network is based on backpropagation and gradient descent. In training process first, weights of edges are initialized. Then input is given to graph network. Output of graph network (result predicted by graph network) is compared to real value of output as in training, we know what actual output is. Error is calculated according to loss function. According to error value, backpropagation happens to update the weight of edges, this process repeated several times to train the graph network. For image recognition process, cross entropy function is taken as loss function. Soft max function is used normalize output values, to get output values in probability range. Training process of graph network can be summarized in seven steps. (1) Randomly initialize the weights to small number close to zero. (2) Input the first observation of dataset in input layer, each feature in one input node. (3) From left to right, the nodes are activated, activation propagates until getting predicted result (output of graph network). (4) Compare the predicted results to actual result to measure error. (5) Backpropagation: from right to left, the error is back propagated. Update the weights according to error. (6) Repeat step 1 to 5 and update the weights after each observation. (7) When whole training set is passed through graph network, that makes an epoch. Do appropriate number of such epochs (on the same training set).

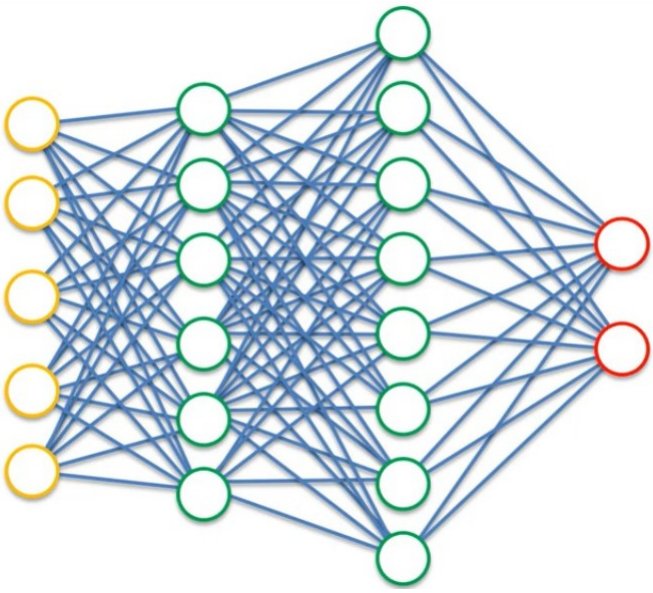

Figure 10: Graph Network
Input layer is shown in yellow, hidden layer in green and output layer in red.

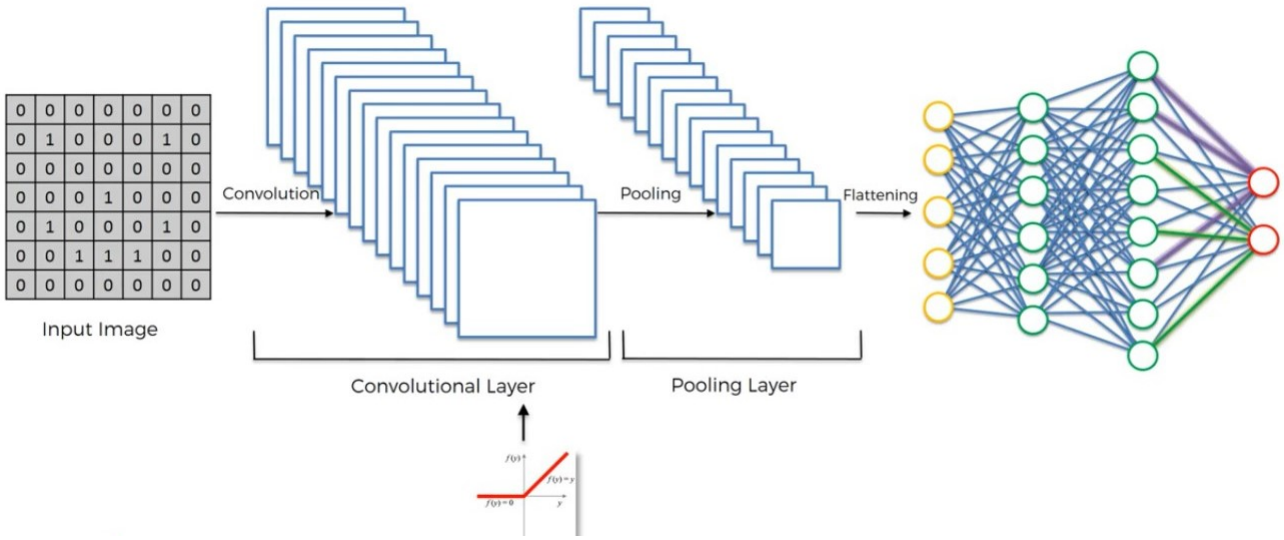

Figure 11: Summary
Source:SlideShare

4. Interpretation and Significance of the Solution for Application

The process and module defined in section 2 and section 3 can be used to write program recognizing image. The program recognizing image can be used in many applications. For example, we can design an application such that if we are traveling in some other country and we see the poster or some information written in language other than we know, we can take out phone and point camera at it, using image recognition - character recognition program we get text in digital format from it and using translator program we can translate it in our preferable language. In food review sites, image recognition can be used to tag the name of food dishes present in image uploaded by the reviewer. Image Recognition can be used to make maps more interactive, we can point our phone camera at shop and using image recognition we can identify the name of shop and using that we can get the opening and closing timings and contact details of that shop. We can also make program for Natural Language Processing using construction similar to Image Recognition technique. For example, we can train the program to tell whether customer liked food or not by processing food review written by customer. In process of recognizing review, we have to use regular expressions instead of matrix operations. In process similar to feature extraction, we have to remove stop words such as 'the', 'is', 'at', 'on', as these words doesn't contribute as a deciding factor or feature. We can convert all other words to lower case and all verbs to present tense. According to this procedure we can get words which are deciding factor for predicting whether customer liked review or not and train our model on these words. I have uploaded that code on website. In this way, we are able to make program which can predict whether customer liked food or not from processing review. We can also make similar program for other problems related to Natural Language Processing. Convolutional Neural Network can be used in video analysis, computer programs which can efficiently play chess or other such games and medical applications. Image Recognition system can also be used in driverless car. Facial recognition system can be made using image recognition program which can be used as face unlock system for mobile phones.

5. Commercialization

Image recognition program can detect objects and people present in photo. Therefore it can associate name of people and objects as tags to that images. With the help of image recognition program tags can be automatically added to photo. We can also add location (place where the photo is taken) in tags. We can use this tags in photo searching process. For example, if we want to get photos of particular person or pet dog or particular object from photo gallery which contains large

amount of photos, then we don't have to scroll manually, we can just type the name of person or object and photos which have that name associated with them as a tag will show up. This process can be further extended, we can search for a photo just by describing the photo in words. For example, search term can be like my photo with my pet dog at riverfront. In this way by using image recognition program and with the help of natural language processing, we are able to make an application that can search for a photo using description in words in photo gallery of phone or from photos on pc or any other large database of photos.

With the help of image recognition program, we are also able to make an app that can convert handwritten text or printed text to digital text format. For example, we can take a photo a of paper, which contains handwritten or printed text and application made using image recognition program can get text in digital format from that photo. Therefore application will make text file from handwritten or printed text in photo.

Image recognition program can be further extended to recognize hand gestures using camera. We can operate pc from distance by hand gestures using hand gesture recognition by camera. Image recognition plays big role in integrating physical world with digital world. We need a camera and a processor to detect hand gesture, we can assign different hand gestures to different commands, we can execute command on pc or phone according to detected hand gesture. There are electric appliances available, which can be turned on and off from giving signal through WiFi using mobile phone application. Therefore we can make a system using camera and a processor for a home that can detect hand gesture and turn on and off electric appliances according to detected hand gesture. Image Recognition also plays big role in Artificial Intelligence.

References

[1] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[2] C-C Jay Kuo. Understanding convolutional neural networks with a mathematical model. *Journal of Visual Communication and Image Representation*, 41:406–413, 2016.