A ROBUST SIGN LANGUAGE AND HAND GESTURE RECOGNITION SYSTEM USING CONVOLUTION NEURAL NETWORKS

PROJECT DOCUMENTATION

submitted by

D Prakhya	316126510168
M Sri Manjari	316126510152
A Varaprasadh	316126510201
NSV Krishna Reddy	316126510163
D Krishna Vamsi	315126510045

in fulfilment for the Major Project of

PROJECT LAB

in

COMPUTER SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGIES AND SCIENCES

SANGIVALASA, BHEEMUNIPATNAM, VISAKHAPATNAM-531 162

2016-2020

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGIES AND SCIENCES

(AUTONOMOUS)

SANGIVALASA, BHEEMUNIPATNAM, VISAKHAPATNAM-531 162

BONAFIED CERTIFICATE

Certified that this project report **"A ROBUST SIGN LANGUAGE AND HAND GESTURE RECOGNITION SYSTEM USING CONVOLUTION NEURAL NETWORKS"**

is the bonafied work of

D Prakhya	316126510168
M Sri Manjari	316126510152
A Varaprasadh	316126510201
NSV Krishna Reddy	316126510163
D Krishna Vamsi	315126510045

SIGNATURE

Dr. R. SIVARANJANI

(HEAD OF THE DEPARTMENT)

COMPUTER SCIENCE AND ENGINEERING

ANITS

SIGNATURE

Mr.K CHANDRA SEKHAR

ASST.PROFESSOR

SUPERVISOR

COMPUTER SCIENCE AND ENGINEERING

ANITS

DECLARATION

We, **Prakhya.D, M.SriManjari, NSVKReddy, A.Varaprasadh, D.KrishnaVamsi** students of fourth year first semester **B.TECH,COMPUTER SCIENCE AND ENGINEERING(AUTONOMOUS),VISAKHAPATNAM**, hereby declare that the major project work and titled **"A ROBUST SIGN LANGUAGE AND HAND GESTURE RECOGNITION SYSTEM USING CONVOLUTION NEURAL NETWORKS"**is carried by us and submitted in fulfilment of the requirements for "MAJOR PROJECT OF PROJECT LAB", under Anil Neerukonda Institute of Technologies &Sciences during the Academic years 2016-2020 and has not been submitted the project to any other university.

D Prakhya	316126510168
M Sri Manjari	316126510152
A Varaprasadh	316126510201
NSV Krishna Reddy	316126510163
D Krishna Vamsi	315126510045

ACKNOWLEDGEMENT

An endeavour over a long period can be successful with the advice and support of many well-wishers. We take this opportunity to express our gratitude and appreciation to all of them.We owe our tributes to **Dr. R. Sivaranjani, Head of the Department, Computer Science & Engineering** for her valuable support and guidance during the period of project implementation.

We wish to express our sincere thanks and gratitude to our project guide **Mr. K. Chandra Sekhar**, **Assistant Professor** ,**Department of Computer Science and Engineering, ANITS**, for the stimulating discussions, in analyzing problems associated with our project work and for guiding us throughout the project. Project meetings were highly informative. We express our warm and sincere thanks for the encouragement, untiring guidance and the confidence they had shown in us. We are immensely indebted for their valuable guidance throughout our project.

We also thank all the staff members of CSE department for their valuable advices.

We also thank Principal and supporting staff for providing resources as and when required.

D Prakhya	316126510168
M Sri Manjari	316126510152
A Varaprasadh	316126510201
NSV Krishna Reddy	316126510163
D Krishna Vamsi	315126510045

TABLE OF CONTENTS

FACULTY NAME: K.Chandra Sekhar

PRIMARY RESEARCH AREA: ImageProcessing

SECONDARY RESEARCH AREA: Deep Neural Network

Project Title : A ROBUST SIGN LANGUAGE AND HAND GESTURE RECOGNITION SYSTEM USING CONVOLUTIN NEURAL NETWORKS

ABSTRACT

Sign language is the only tool of communication for the person who is not able to speak and hear anything. Sign language is a boon for the physically challenged people to express their thoughts and emotion. In this work, a novel scheme of sign language recognition has been proposed for identifying the alphabets and gestures in sign language. With the help of computer vision and neural networks we can detect the signs and give the respective text output.

1.INTRODUCTION

1.1 PROBLEM STATEMENT

Speech impaired people use hand signs and gestures to communicate. Normal people face difficulty in understanding their language. Hence there is a need of a system which recognizes the different signs, gestures and conveys the information to the normal people. It bridges the gap between physically challenged people and normal people.

1.2 IMAGE PROCESSING

Image processing is a method to perform some operations on an image, in order to get an enhanced image or to extract some useful information from it. It is a type of signal processing in which input is an image and output may be image or characteristics/features associated with that image. Nowadays, image processing is among rapidly growing technologies. It forms core research area within engineering and computer science disciplines too.

Image processing basically includes the following three steps:

- Importing the image via image acquisition tools.
- Analysing and manipulating the image.
- Output in which result can be altered image or report that is based on image analysis.

1.3 SIGN LANGUAGE

A language that employs signs made with the hands and other movements, including facial expressions and postures of the body, used primarily by people who are deaf. There are many different sign languages as, for example, British and American sign languages. British sign language (BSL) is not easily intelligible to users of American sign Language (ASL). Unlike ASL, BSL uses a two-handed alphabet. In developing countries, deaf people may use the sign language of educators and missionaries from elsewhere in the world.

1.4 SEGMENTATION

Segmentation is the process of partitioning a digital image into multiple segments. The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics.

The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours extracted from the image . Each of the pixels in a region are similar with respect to some characteristic or computed property, such ascolor, intensity, or texture. Adjacent regions are significantly different with respect to the same characteristic. When applied to a stack of images, typical in medical imagining, the resulting contours after image segmentation can be used to create 3D reconstructions with the help of interpolation algorithms like marching cubes.

Image segmentation is the process of partitioning a digital image into multiple segments(sets of pixels, also known as image objects). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyse.Modern image segmentation techniques are powered by deep learning technology.

1.5 STEPS FOR PRE-PROCESSING

There are 3 important steps in image preprocessing. They are:

1.5.1 Uniform aspect ratio:

An *aspect ratio* is a proportional relationship between an image's width and height. Essentially, it describes an image's shape. Since images don't need to have the same dimensions to have the same *aspect ratio*, it's better to crop them to a specific ratio than to try to match their exact dimensions

1.5.2 Image Scaling:

Once we've ensured that all images are square (or have some predetermined aspect ratio), it's time to scale each image appropriately. We've decided to have images with width and height of 100 pixels. We'll need to scale the width and height of each image by a factor of 0.4 (100/250). There are a wide variety of up-scaling and down-scaling techniques and we usually use a library function to do this for us.

1.5.3 Data augmentation:

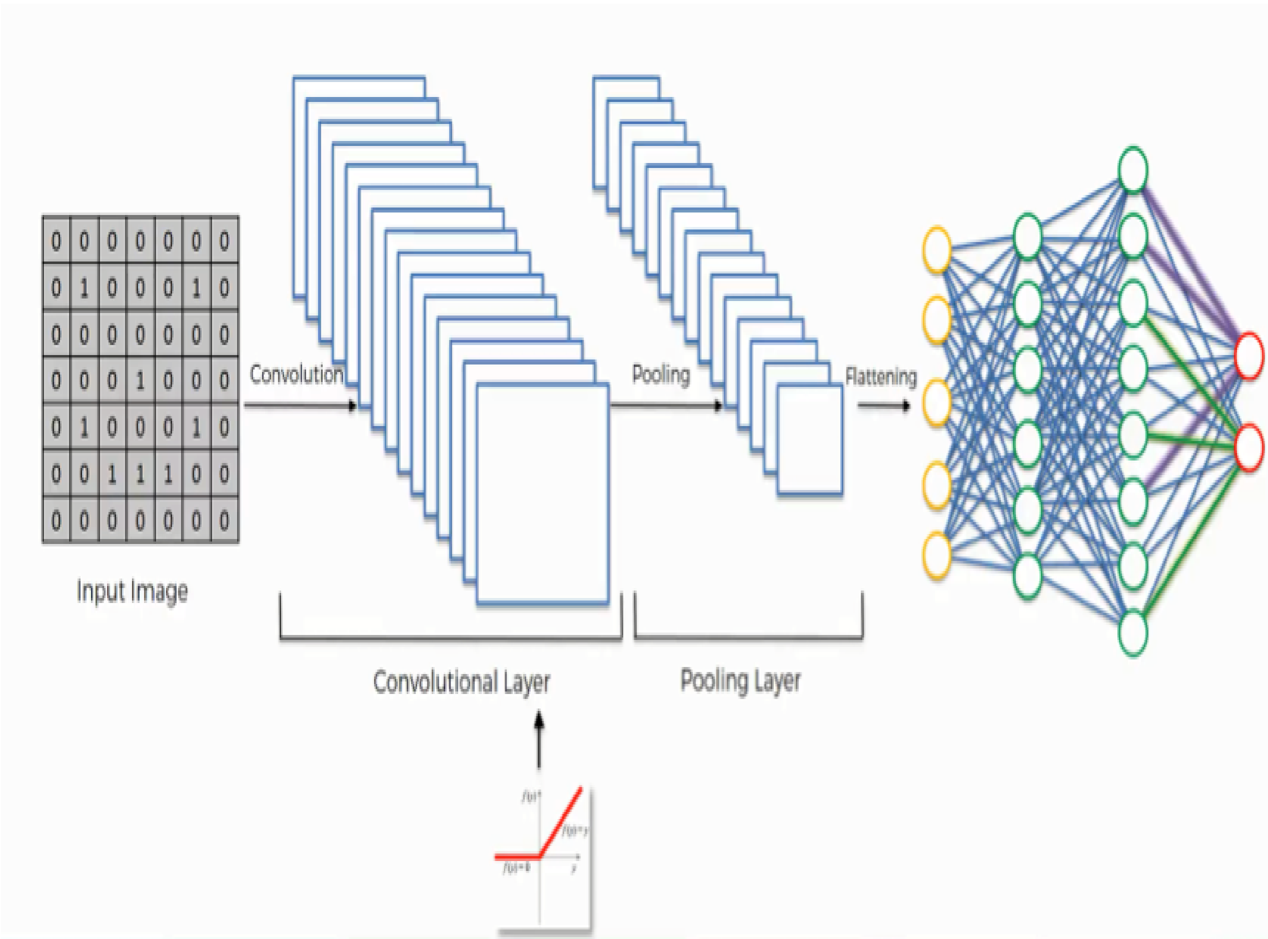
Another common pre-processing technique involves augmenting the existing data-set with perturbed versions of the existing images. Scaling, rotations and other affine transformations are typical. This is done to expose the neural network to a wide variety of variations. This makes it less likely that the neural network recognizes unwanted characteristics in the data-set.

1.6 CONVOLUTION NEURAL NETWORK

Convolutional neural networks (CNN) is a special architecture of artificial neural networks, proposed by Yann LeCun in 1988. CNN uses some features of the visual cortex. One of the most popular uses of this architecture is image classification. For example Facebook uses CNN for automatic tagging algorithms, Amazon — for generating product recommendations and Google — for search through among users' photos.

Instead of the image, the computer sees an array of pixels. For example, if image size is 300 x 300. In this case, the size of the array will be 300x300x3. Where 300 is width, next 300 is height and 3 is RGB channel values. The computer is assigned a value from 0 to 255 to each of these numbers. This value describes the intensity of the pixel at each point.

To solve this problem the computer looks for the characteristics of the baselevel. In human understanding such characteristics are for example the trunk or large ears. For the computer, these characteristics are boundaries or curvatures. And then through the groups of convolutional layers the computer constructs more abstract concepts. In more detail: the image is passed through a series of convolutional, nonlinear, pooling layers and fully connected layers, and then generates the output.



2.LITERATURE SURVEY

2.1 INTRODUCTION:

In Literature survey we have gone through other similar works that are implemented in the domain of sign language recognition.The summaries of each of the project works are mentioned in the following chapter.

2.2 EXISTING MODELS:

1)A Survey of Hand Gesture Recognition Methods in Sign Language Recognition

The studies are based on various input sensors, gesture segmentation, extraction of features and classification methods. This paper aims to analyze and compare the methods employed in the SLR systems, classifications methods that have been used, and suggests the most promising method for future research. Due to recent advancement in classification methods, many of the recent proposed works mainly contribute on the classification methods, such as hybrid method and Deep Learning. This paper focuses on the classification methods used in prior Sign Language Recognition system. Based on our review, HMM-based approaches have been explored extensively in prior research, including its modifications.

Hybrid CNN-HMM and fully Deep Learning approaches have shown promising results and offer opportunities for further exploration.

2)Communication between Deaf-Dumb People and Normal People

Chat applications have become a powerful media that assist people to communicate in different languages with each other. There are lots of chat applications that are used different people in different languages but there are not such a chat application that has facilitate to communicate with sign languages. Sign languages are used by deaf and dump people to communicate among them but those Sign languages vary from nation to nation as American Sign Language, British Sign language, Japanese Sign language etc. The developed system has based on Sinhala Sign language. The system has included four main components as text messages are converted to signmessages, voice messages are converted to sign messages, sign messages are converted to text messages and sign messages are converted to voice messages. Google voice recognition API has used to develop speech character recognition for voice messages. The system has trained for the speech and text patterns by using some text parameters and Signs of Sinhala Sign language is displayed by emoji. Those emoji and signs that are included in this system will pave a new way for the normal people to be more close to hearing disable people and also hearing disable people to be more close to normal people

3)A System for Recognition of Indian Sign Language for Deaf People using Otsu's Algorithm

In this paper we proposed some methods, through which the recognition of the signs becomes easy for peoples while communication. We use the different symbols of signs to convey the meanings. And the result of those symbols signs will be converted into the text. In this project, we are capturing hand gestures through webcam and convert this image into gray scale image. The segmentation of gray scale image of a hand gesture is performed using Otsu thresholding algorithm.. Total image level is divided into two classes one is hand and other is background. The optimal threshold value is determined by computing the ratio between class variance and total class variance. To find the boundary of hand gesture in image Canny edge detection technique is used.

3. METHODOLOGY

3.1 TRAINING MODULE:

Supervised machine learning:It is one of the ways of machine learning where the model is trained by input data and expected output data.

To create such model, it is necessary to go through the following phases:

1. model construction
2. model training
3. model testing
4. model evaluation

Model construction depends on machine learning algorithms. In this projects case, it was neural networks.

Such an algorithm looks like:

1. begin with its object: model = Sequential()
2. then consist of layers with their types: model.add(*type_of_layer()*)
3. after adding a sufficient number of layers the model is compiled. At this moment Keras communicates with TensorFlow for construction of the model. During model compilation it is important to write a loss function and an optimizer algorithm. It looks like: model.comile(loss= 'name_of_loss_function', optimizer= 'name_of_opimazer_alg') The loss function shows the accuracy of each prediction made by the model.

Before model training it is important to scale data for their further use.

Model training:

After model construction it is time for **model training.** In this phase, the model is trained using training data and expected output for this data.

It’s look this way: model.fit(training_data, expected_output).

Progress is visible on the console when the script runs. At the end it will report the final accuracy of the model.

Model Testing:

During this phase a second set of data is loaded. This data set has never been seen by the model and therefore it’s true accuracy will be verified.

After the model training is complete, and it is understood that the model shows the right result, it can be saved by: model.save(“name_of_file.h5”).

Finally, the saved model can be used in the real world. The name of this phase is **model evaluation**. This means that the model can be used to evaluate new data.

3.2 SEGMENTATION:

Image segmentation is the process of partitioning a digital image into multiple segments

(sets of pixels, also known as image objects). The goal of segmentation is to simplify and/or

change the representation of an image into something that is more meaningful and easier to analyse.

Modern image segmentation techniques are powered by deep learning technology. Here are several deep learning architectures used for segmentation:

Why does Image Segmentation even matter?

If we take an example of Autonomous Vehicles, they need sensory input devices like cameras, radar, and lasers

to allow the car to perceive the world around it, creating a digital map. Autonomous driving is not even possible

without object detection which itself involves image classification/segmentation.

How Image Segmentation works

Image Segmentation involves converting an image into a collection of regions of pixels that are represented by a mask or a labeled image. By dividing an image into segments, you can process only the important segments of the image instead of processing the entire image.

A common technique is to look for abrupt discontinuities in pixel values, which typically indicate edges that define a region.Another common approach is to detect similarities in the regions of an image. Some techniques that follow this approach are region growing, clustering, and thresholding.

A variety of other approaches to perform image segmentation have been developed over the years using domain-specific knowledge to effectively solve segmentation problems in specific application areas.

The Image segmentation in our project is done in 3 steps:

1)Edge detection

2)Masking

3)Clustering

3.2.1 EDGE DETECTION:

Cannys edge detection was preferred among the others

Gradient Calculation

The Gradient calculation step detects the edge intensity and direction by calculating the gradient of the image using edge detection operators.

Edges correspond to a change of pixels' intensity. To detect it, the easiest way is to apply filters that highlight this intensity change in both directions: horizontal (x) and vertical (y)

When the image is smoothed, the derivatives Ix and Iy w.r.t. x and y are calculated. It can be implemented by convolving I with Sobel kernels Kx and Ky, respectively:

Sobel filters for both direction (horizontal and vertical)

Then, the magnitude G and the slope θ of the gradient are calculated as follow:

Gradient intensity and Edge direction

Double threshold:

The double threshold step aims at identifying 3 kinds of pixels: strong, weak, and non-relevant:

Strong pixels are pixels that have an intensity so high that we are sure they contribute to the final edge.

Weak pixels are pixels that have an intensity value that is not enough to be considered as strong ones, but yet not small enough to be considered as non-relevant for the edge detection. Other pixels are considered as non-relevant for the edge.

3.2.3 CLUSTERING:

So let us start with one of the clustering-based approaches in Image Segmentation which is K-Means clustering.

K-Means clustering algorithm:

K-Means clustering algorithm is an unsupervised algorithm and it is used to segment the interest area from the background.

It clusters, or partitions the given data into K-clusters or parts based on the K-centroids.

The algorithm is used when you have unlabeled data(i.e. data without defined categories or groups). The goal is to find certain

groups based on some kind of similarity in the data with the number of groups represented by K.

The objective of K-Means clustering is to minimize the sum of squared distances between all points and the cluster center.

Steps in K-Means algorithm:

1. Choose the number of clusters K.

2. Select at random K points, the centroids(not necessarily from your dataset).

3. Assign each data point to the closest centroid → that forms K clusters.

4. Compute and place the new centroid of each cluster.

5. Reassign each data point to the new closest centroid.

   If any reassignment . took place, go to step 4, otherwise, the model is ready.

How to choose the optimal value of K?

For a certain class of clustering algorithms (in particular K-Means, K-medoids, and expectation-maximization algorithm),

there is a parameter commonly referred to as K that specifies the number of clusters to detect.

Other algorithms such as DBSCAN and OPTICS algorithm do not require the specification of this parameter;

Hierarchical Clustering avoids the problem altogether but that's beyond the scope of this article.

If we talk about K-Means then the correct choice of K is often ambiguous, with interpretations depending on the shape and scale of the distribution of points in a data set and the desired clustering resolution of the user. In addition, increasing K without penalty will always reduce the amount of error in the resulting clustering, to the extreme case of zero error if each data point is considered its own cluster (i.e., when K equals the number of data points, n). Intuitively then, the optimal choice of K will strike a balance between maximum compression of the data using a single cluster, and maximum accuracy by assigning each data point to its own cluster.

If an appropriate value of K is not apparent from prior knowledge of the properties of the data set, it must be chosen somehow. There are several categories of methods for making this decision and Elbow method is one such method.

3.3 IMAGE PRE-PROCESSING:

3.3.1 UNIFORM ASPECT RATIO

Understanding aspect ratios:

An *aspect ratio* is a proportional relationship between an image's width and height. Essentially, it describes an image's shape.

Aspect ratios are written as a formula of width to height, like this: **3:2**.

For example, a square image has an *aspect ratio* of 1:1, since the height and width are the same. The image could be 500px × 500px, or 1500px × 1500px, and the *aspect ratio* would still be 1:1.

As another example, a portrait-style image might have a ratio of 2:3. With this *aspect ratio*, the height is 1.5 times longer than the width. So the image could be 500px × 750px, 1500px × 2250px, etc.

Cropping to an *aspect ratio*

Aside from using built-in Site Styles options, you may want to manually crop an image to a certain *aspect ratio*. For example, if you use product images that have same *aspect ratio*, they'll all crop the same way on your site.

Option 1 - Crop to a pre-set shape

Use the built-in Image Editor to crop images to a specific shape. After opening the editor, use the Crop tool to choose from preset *aspect ratio*s.

Option 2 - Custom dimensions

To crop images to a custom *aspect ratio* not offered by our built-in Image Editor, use a third-party editor.

Since images don't need to have the same dimensions to have the same *aspect ratio*, it's better to crop them to a specific ratio than to try to match their exact dimensions. For best results, crop the shorter side based on the longer side.

- For instance, if your image is 1500px × 1200px, and you want an *aspect ratio* of 3:1, crop the shorter side to make the image 1500px × 500px.
- Don't scale up the longer side; this can make your image blurry.

3.3.2 IMAGE SCALING:

- In computer graphics and digital imaging, **image scaling** refers to the resizing of a digital image. In video technology, the magnification of digital material is known as upscaling or resolution enhancement.
- When scaling a vector graphic image, the graphic primitives that make up the image can be scaled using geometric transformations, with no loss of image quality. When scaling a raster graphics image, a new image with a higher or lower number of pixels must be generated. In the case of decreasing the pixel number (scaling down) this usually results in a visible quality loss. From the standpoint of digital signal processing, the scaling of raster graphics is a two-dimensional example of sample-rate conversion, the conversion of a discrete signal from a sampling rate (in this case the local sampling rate) to another.

Algorithms

Nearest-neighbor interpolation:

One of the simpler ways of increasing image size is nearest-neighbor interpolation, replacing every pixel with the nearest pixel in the output; for upscaling this means multiple pixels of the same color will be present. This can preserve sharp details in pixel art, but also introduce jaggedness in previously smooth images. 'Nearest' in nearest-neighbor doesn't have to be the mathematical nearest. One common implementation is to always round towards zero. Rounding this way produces fewer artifacts and is faster to calculate.

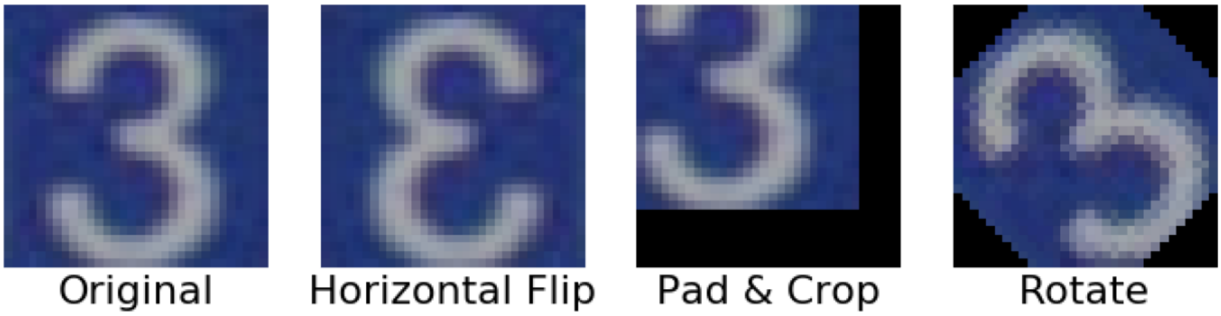
Mipmap:

Another solution to the downscale problem of bi-sampling scaling are mipmaps. A mipmap is a prescaled set of downscale copies. When downscaling the nearest larger mipmap is used as the origin, to ensure no scaling below the useful threshold of bilinear scaling is used. This algorithm is fast, and easy to optimize. It is standard in many frameworks such as OpenGL. The cost is using more image memory, exactly one third more in the standard implementation.

Deep convolutional neural networks:
This method uses machine learning for more detailed images such as photographs and complex artwork. Programs that use this method include waifu2x and Neural Enhance.

3.2.3 DATA AUGMENTATION

Data augmentation is a strategy that enables practitioners to significantly increase the diversity of data available for training models, without actually collecting new data. Data augmentation techniques such as cropping, padding, and horizontal flipping are commonly used to train large neural networks. However, most approaches used in training neural networks only use basic types of augmentation. While neural network architectures have been investigated in depth, less focus has been put into discovering strong types of data augmentation and data augmentation policies that capture data invariances.



An image of the number "3" in original form and with basic augmentations applied.

3.4 Classification: Convolution Neural network

Image classification is the process of taking an input(like a picture) and outputting its class or probability that the input is a particular class.

A convolutional neural network convolves learned featured with input data and uses 2D convolution layers.

3.4.1 CONVOLUTION OPERATION:

In purely mathematical terms, convolution is a function derived from two given functions by integration which expresses how the shape of one is modified by the other.

Convolution formula:

$$(f * g)(t) \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} f(\tau)\, g(t-\tau)\, d\tau$$

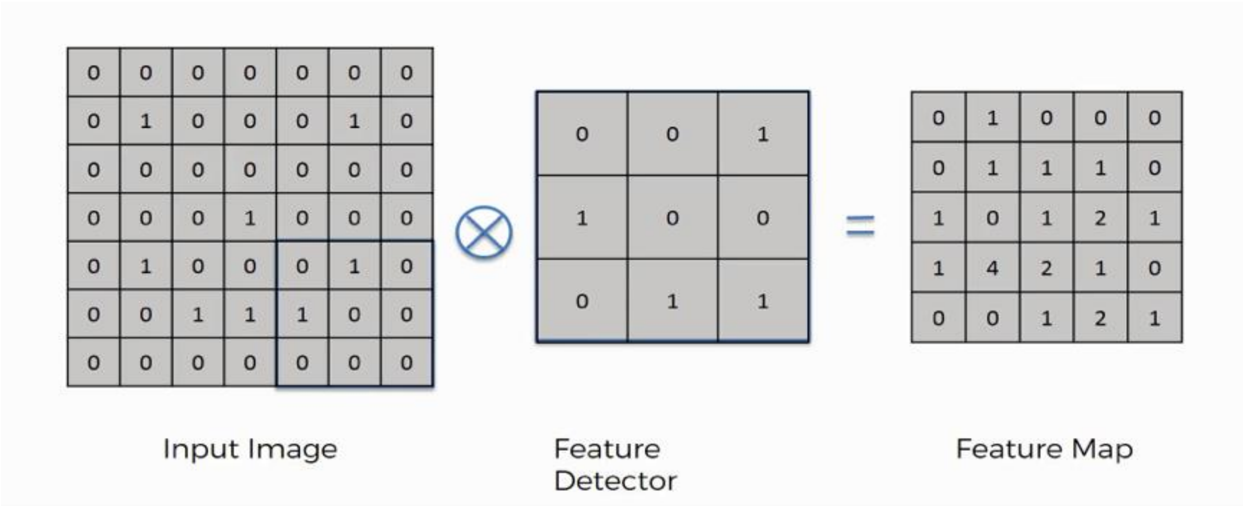
Here are the three elements that enter into the convolution operation:

- Input image
- Feature detector
- Feature map

Steps to apply convolution layer:

- You place it over the input image beginning from the top-left corner within the borders you see demarcated above, and then you count the number of cells in which the feature detector matches the input image.

- The number of matching cells is then inserted in the top-left cell of the feature map.
- You then move the feature detector one cell to the right and do the same thing. This movement is called a and since we are moving the feature detector one cell at time, that would be called a stride of one pixel.
- What you will find in this example is that the feature detector's middle-left cell with the number 1 inside it matches the cell that it is standing over inside the input image. That's the only matching cell, and so you write "1" in the next cell in the feature map, and so on and so forth.
- After you have gone through the whole first row, you can then move it over to the next row and go through the same process.



There are several uses that we gain from deriving a feature map. These are the most important of them: Reducing the size of the input image, and you should know that the larger your strides (the movements across pixels), the smaller your feature map.

3.4.2 RELU LAYER:

Rectified linear unit is used to scale the parameters to non negative values.We get pixel values as negative values too . Inthis layer we make them as 0's.

The purpose of applying the rectifier function is to increase the non-linearity in our images. The reason we want to do that is that images are naturally non-linear.
The rectifier serves to break up the linearity even further in order to make up for the linearity that we might impose an image when we put it through the convolution operation.

What the rectifier function does to an image like this is remove all the black elements from it, keeping only those carrying a positive value (the grey and white colors).The essential difference between the non-rectified version of the image and the rectified one is the progression of colors. After we rectify the image, you will find the colors changing more abruptly. The gradual change is no longer there. That indicates that the linearity has been disposed of.

3.4.3 POOLING LAYER:

The pooling (POOL) layer reduces the height and width of the input. It helps reduce computation, as well as helps make feature detectors more invariant to its position in the input

This process is what provides the convolutional neural network with the “spatial variance” capability. In addition to that, pooling serves to minimize the size of the images as well as the number of parameters which, in turn, prevents an issue of “overfitting” from coming up. Overfitting in a nutshell is when you create an excessively complex model in order to account for the idiosyncracies we just mentioned
The result of using a pooling layer and creating down sampled or pooled feature maps is a summarized version of the features detected in the input. They are useful as small changes in the location of the feature in the input detected by the convolutional layer will result in a pooled feature map with the feature in the same location. This capability added by pooling is called the model’s invariance to local translation.

3.4.4 FULLY CONNECTED LAYER:

The role of the artificial neural network is to take this data and combine the features into a wider variety of attributes that make the convolutional network more capable of classifying images, which is the whole purpose from creating a convolutional neural network.
It has neurons linked to each other ,and activates if it identifies patterns and sends signals to output layer .the outputlayer gives output class based on weight values.

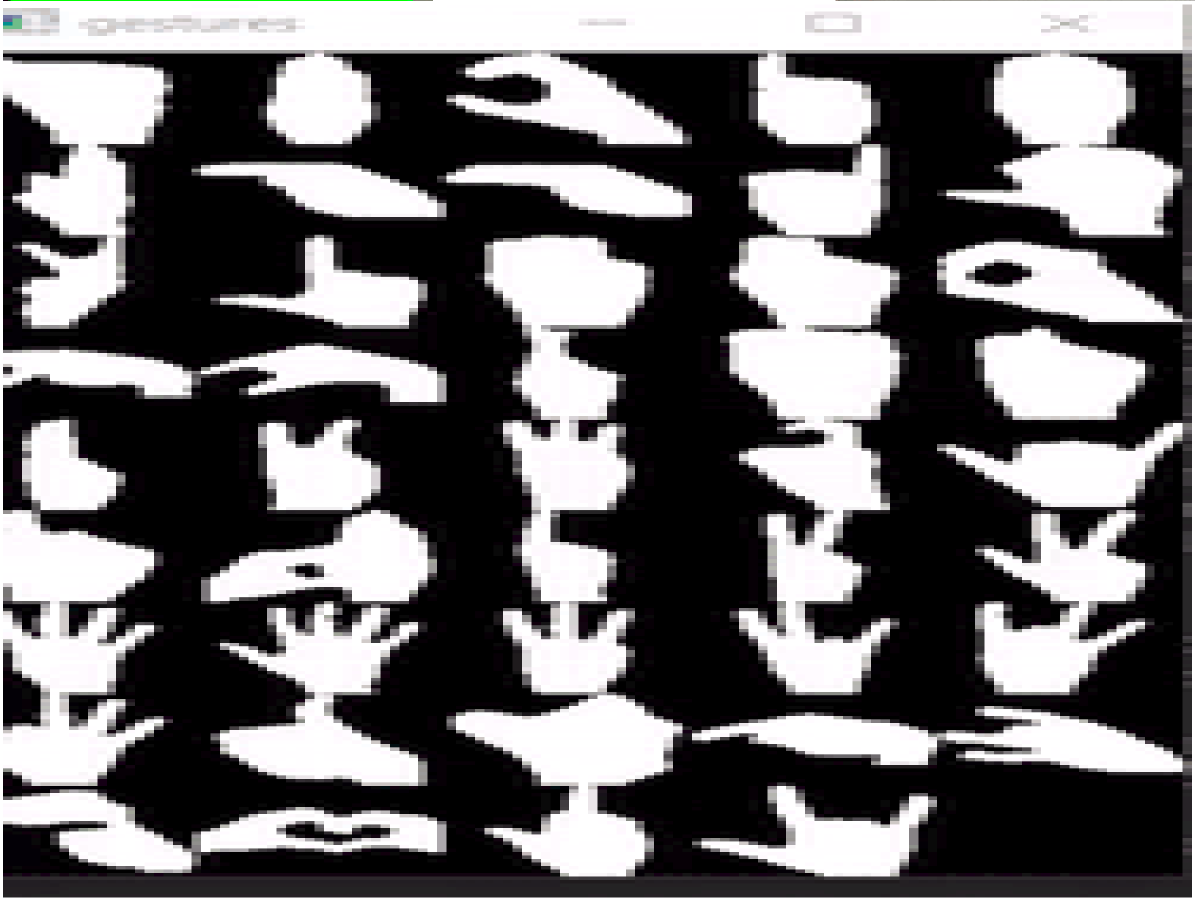
For now, all you need to know is that the loss function informs us of how accurate our network is, which we then use in optimizing our network in order to increase its effectiveness. That requires certain things to be altered in our network.

These include the weights (the blue lines connecting the neurons, which are basically the synapses), and the feature detector since the network often turns out to be looking for the wrong features and has to be reviewed multiple times for the sake of optimization.

This full connection process practically works as follows:

- The neuron in the fully-connected layer detects a certain feature; say, a nose.
- It preserves its value.
- It communicates this value to the classes trained images.

4.EXPECTED INPUT AND OUTPUT:



OUTPUT: