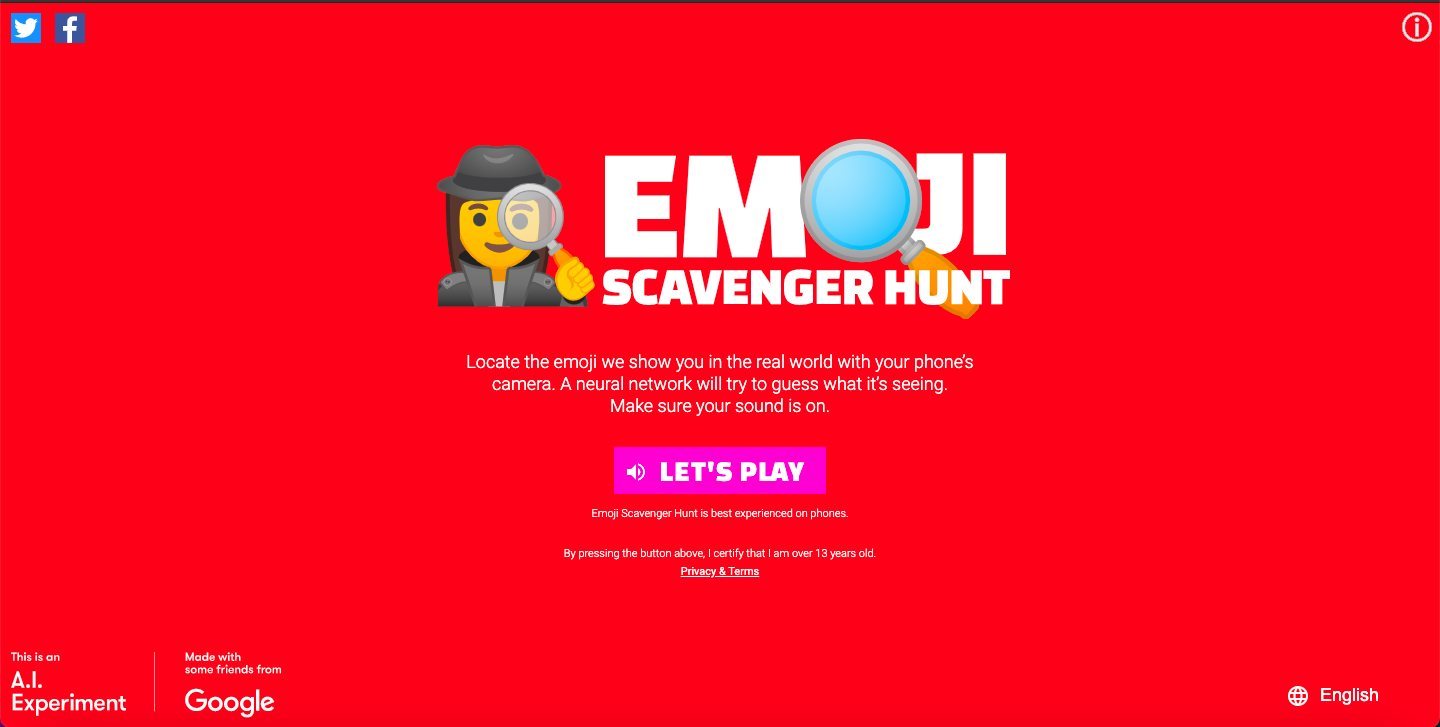
COMPUTER VISION

* COMPUTER VISION INTRODUCTION

As we all know, artificial intelligence is a technique that enables computers to mimic human intelligence. As humans we can see things, analyse it and then do the required action on the basis of what we see. But can machines do the same? Can machines have the eyes that humans have? If you answered Yes, then you are absolutely right. The Computer Vision domain of Artificial Intelligence, enables machines to see through images or visual data, process and analyse them on the basis of algorithms and methods in order to analyse actual phenomena with images. Now before we get into the concepts of Computer Vision, let us experience this domain with the help of the following game:

Emoji Scavenger Hunt : <https://emojiscavengerhunt.withgoogle.com/>

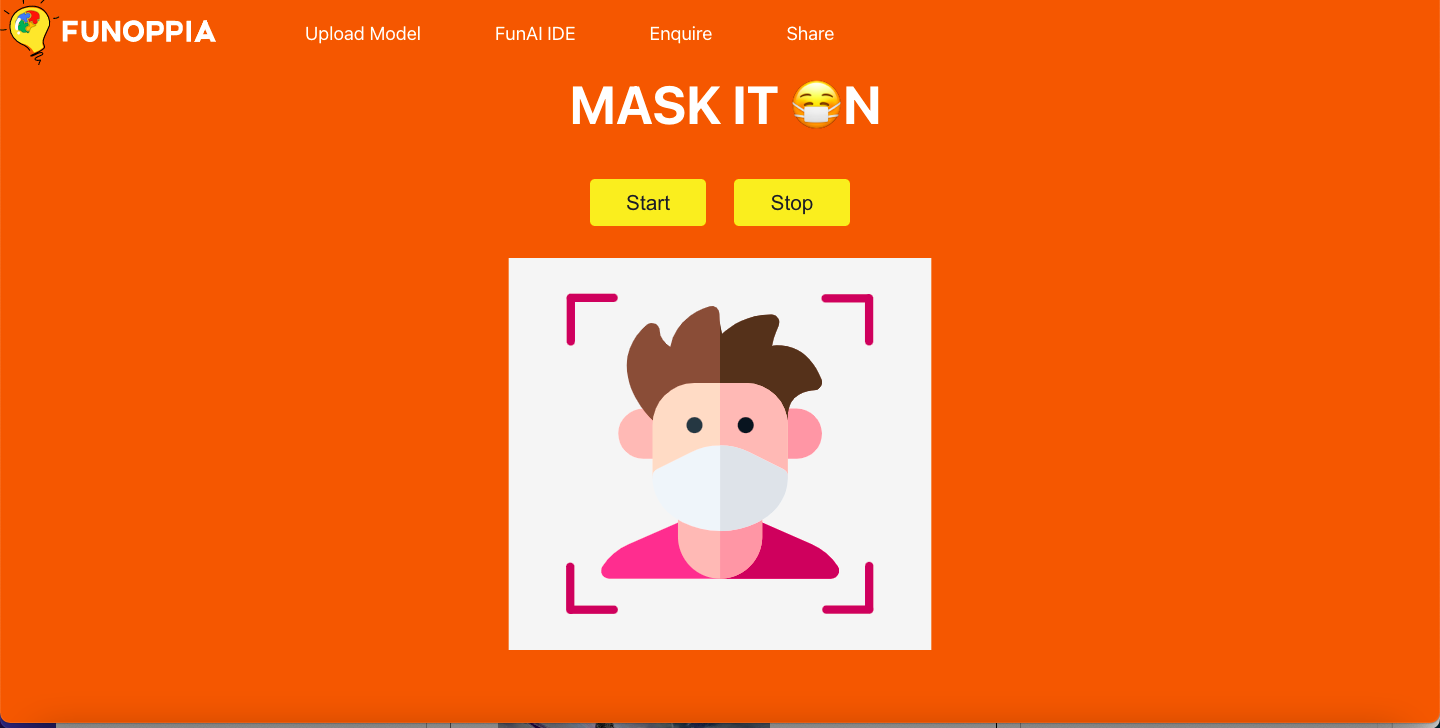


Go to the link and try to play the game of Emoji Scavenger Hunt. The challenge here is to find 8 items within the time limit to pass.

* APPLICATIONS OF CV

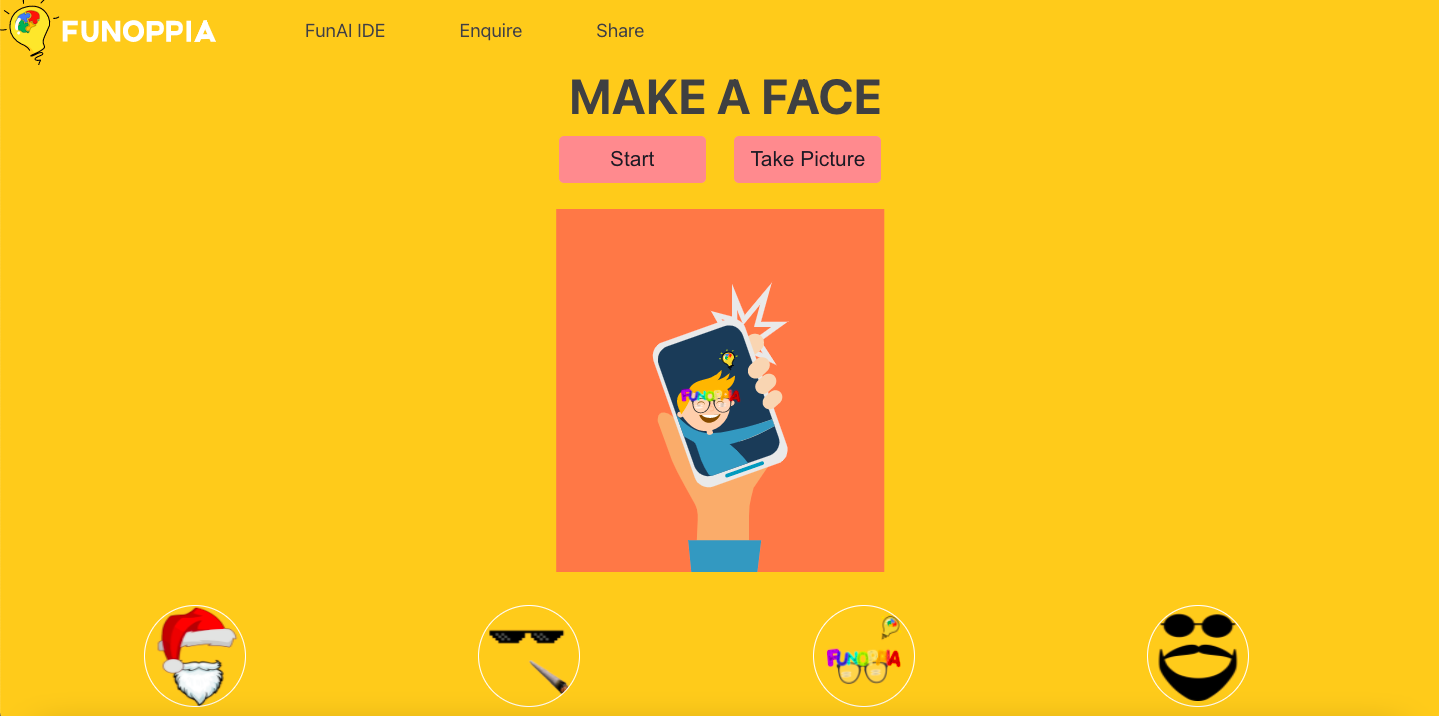
1. Face Mask Recognition Software –

This project checks whether the person using it is wearing a mask or not, go try and check it out on - <https://projects.funoppia.com/mask/>



1. Make a Face (Filters) -

This is a project which puts filters on your face to try on. Try the project on -<https://projects.funoppia.com/filter/>



1. Self Driving Cars –

Computer Vision is the fundamental technology behind developing autonomous vehicles. Most leading car manufacturers in the world are reaping the benefits of investing in artificial intelligence for developing on-road versions of hands-free technology. This involves the process of identifying the objects, getting navigational routes and also at the same time environment monitoring.



1. Google’s Search by Image -

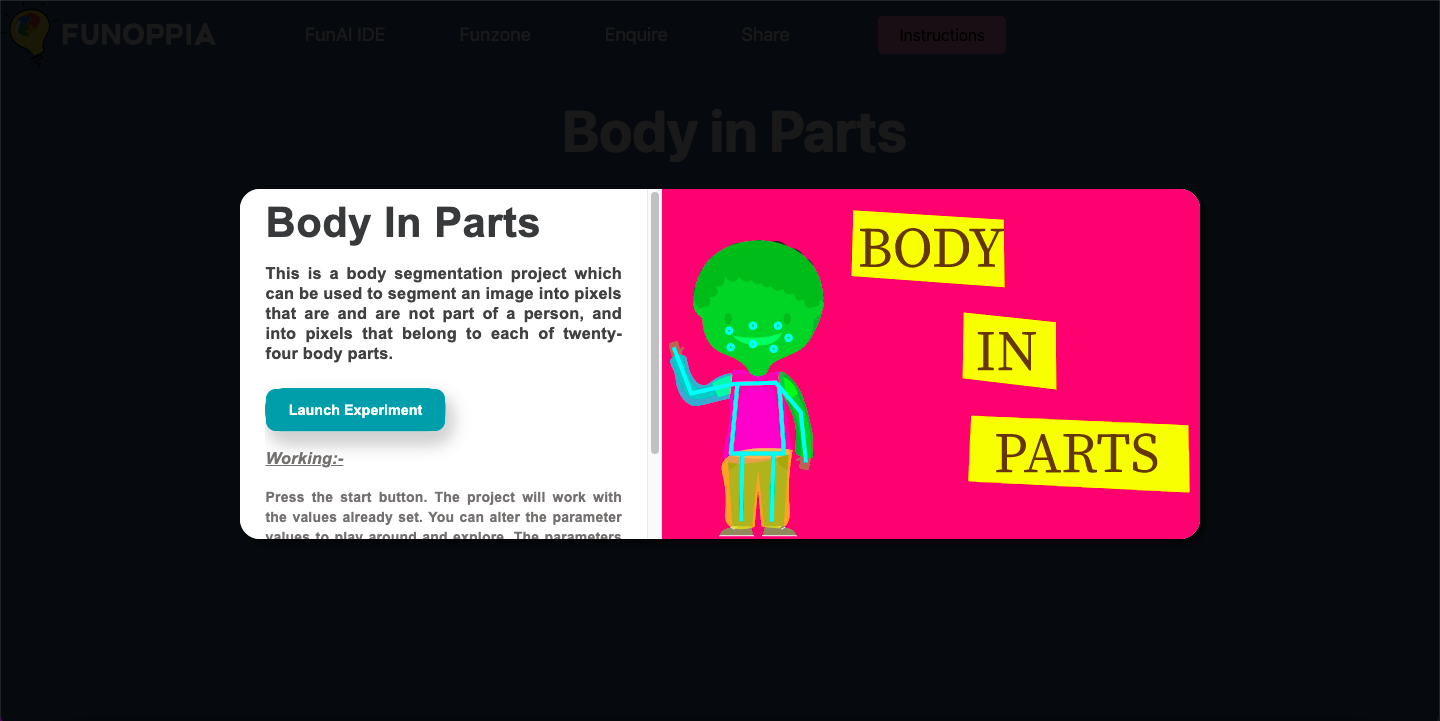


The maximum amount of searching for data on Google’s search engine comes from textual data, but at the same time it has an interesting feature of getting search results through an image. This uses Computer Vision as it compares different features of the input image to the database of images and give us the search result while at the same time analysing various features of the image

.

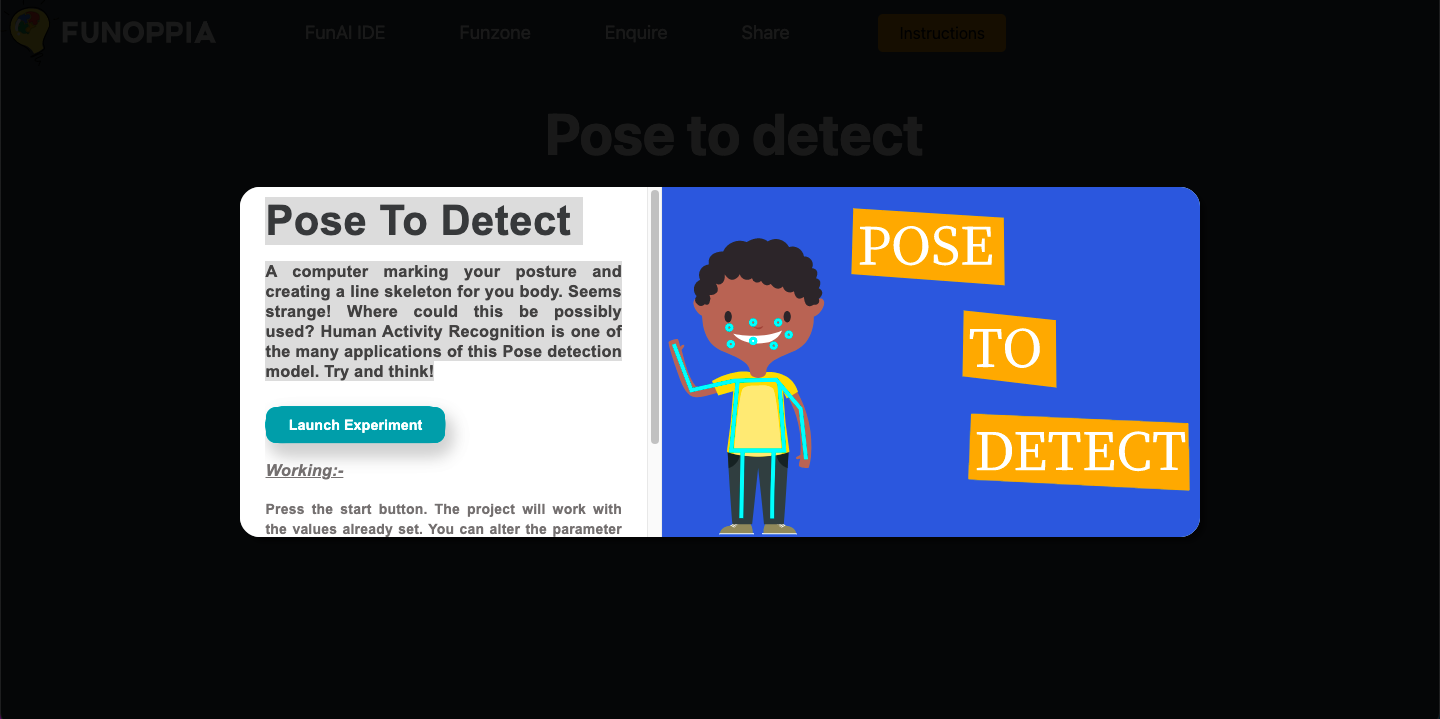
1. Body In Parts

This is a body segmentation project which can be used to segment an image into pixels that are and are not part of a person, and into pixels that belong to each of twenty-four body parts. - <https://projects.funoppia.com/body/>



1. Pose To Detect

A computer marking your posture and creating a line skeleton for you body. Seems strange! Where could this be possibly used? Human Activity Recognition is one of the many applications of this Pose detection model. Try and think! <https://projects.funoppia.com/pose/>



# **PIXEL**

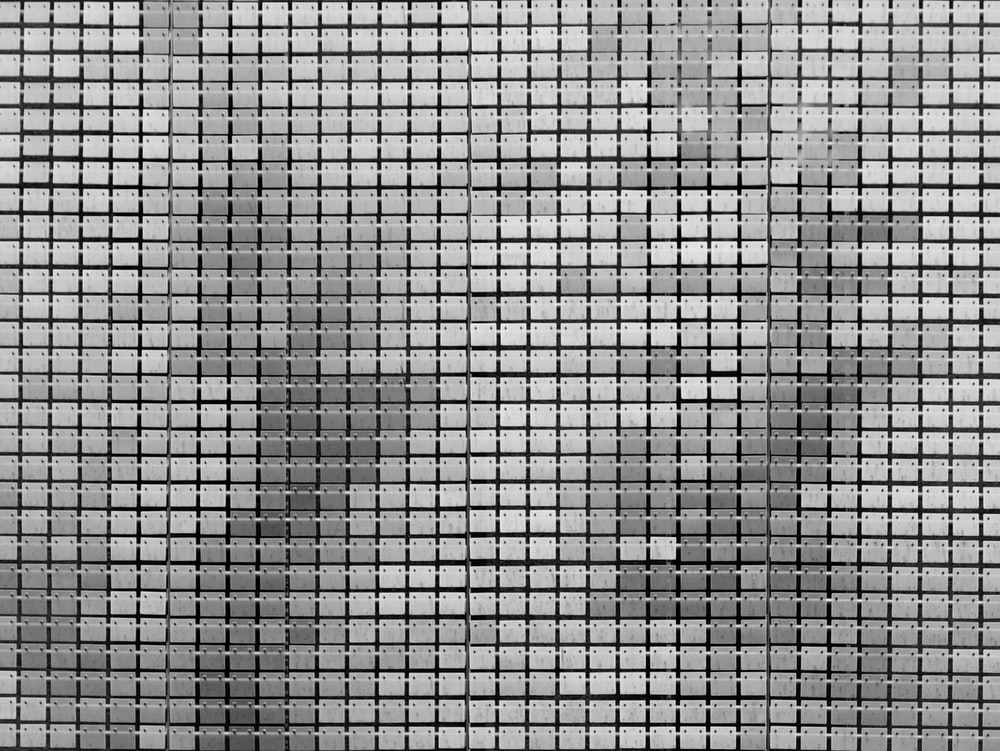
A pixel, or picture element, in the context of computer vision, is the numerical value of the scalar (gray scale or index) or vector (color or multispectral) information at one point in a picture, or image. An image is typically represented as an array of pixels. View different pixels here by zooming in and zooming out of the image

<https://csfieldguide.org.nz/en/interactives/pixel-viewer/>



* HOW DO COMPUTERS SEE IMAGES?

How does our brain work? If our goal is to emulate human vision, CV should be achieved in a manner similar to how our brain enables vision. Of course, we do not yet fully understand how our brain works.  
How do we achieve it? A computer sees an image as series of pixels with it transforms to and array. The image below shows a simplified example of the process.



In this example, each pixel is represented by a number from 0-255 (RGB color code). For a 12×16 image size, we end up with an array of 12 x 16 8-bit integer values. If instead of black and white, the image was in full color for each pixel, we would have three values.

* RESOLUTION

The number of pixels in an image is sometimes called the resolution. When the term is used to describe pixel count, one convention is to express resolution as the width by the height, for example a monitor resolution of 1280×1024. This means there are 1280 pixels from one side to the other, and 1024 from top to bottom.

# **IMAGE FEATURES FOR CLASSIFICATION**

**Classification problems in image and signal analysis** require, on the algorithmic side, to take into account complex information embedded in the data. Images might contain many thousands of pixel values in several colour channels; their correlation and relationship characterizes the class and enables drawing a separation criteria from other classes. It is generally non-feasible to integrate all this information in reasonable running time for classification problems. Therefore, image and signal features are extracted as a representatives of each object and its class.

* INTRODUCTION TO OPENCV

OpenCV or Open Source Computer Vision Library is that tool which helps a computer extract these features from the images. It is used for all kinds of images and video processing and analysis. It is capable of processing images and videos to identify objects, faces, or even handwriting. In this chapter we will use OpenCV for basic image processing operations on images such as resizing, cropping and many more. To install OpenCV library, open command prompt and then write the following command:

**pip install opencv-python**



* CONVOLUTION

We have learnt that computers store images in numbers, and that pixels are arranged in a particular manner to create the picture we can recognize. These pixels have value varying from 0 to 255 and the value of the pixel determines the colour of that pixel. But what if we edit these numbers, will it bring a change to the image? The answer is yes. As we change the values of these pixels, the image changes. This process of changing pixel values is the base of image editing. We all use a lot of image editing software like photoshop and at the same time use apps like Instagram and snapchat, which apply filters to the image to enhance the quality of that image.

As you can see, different filters applied to an image change the pixel values evenly throughout the image. How does this happen? This is done with the help of the process of convolution and the convolution operator which is commonly used to create these effects. Before we understand how the convolution operation works, let us try and create a theory for the convolution operator by experiencing it using an online application.

**TASK**

Go to the link <http://matlabtricks.com/post-5/3x3-convolution-kernels-with-online-demo> and at the bottom of the page click on load “Click to Load Application

* CONVULATION NEURAL NETWORKS (CNN)

Convolutional Neural Network (CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The process of deploying a CNN is as follows:

OUTPUT VALUES

PROCESS IMAGE

INPUT IMAGE



|  |  |
| --- | --- |
| CAR | 75% |
| BIKE | 15% |
| TRUCK | 10% |

Convolutional

Neural Network

* KERNEL

A Kernel is a matrix, which is slid across the image and multiplied with the input such that the output is enhanced in a certain desirable manner. Each kernel has a different value for different kind of effects that we want to apply to an image. In Image processing, we use the convolution operation to extract the features from the images which can le later used for further processing especially in Convolution Neural Network (CNN). In this process, we overlap the centre of the image with the centre of the kernel to obtain the convolution output. In the process of doing it, the output image becomes smaller as the overlapping is done at the edge row and column of the image.

|  |  |  |
| --- | --- | --- |
| -1 | 0 | 0 |
| -1 | 5 | 1 |
| -1 | 0 | 1 |

|  |  |  |
| --- | --- | --- |
| 2 | 2 | 2 |
| 2 | 3 | 2 |
| 1 | 1 | 1 |

13

The middle elements are multiplied and the rest corresponding elements are also multiplied and added to the middle elements.

(5\*3)+(-1\*2)+(0\*2)+(0\*2)+(-1\*2)+(1\*2)+(-1\*1)+(0\*1)+(1\*1)

15 + -2 + 0 + 0 + -2 + 2 + -1 + 0 + 1

= 13.

A convolutional neural network consists of the following layers:

1) Convolution Layer

2) Rectified linear Unit (ReLU)

3) Pooling Layer

4) Fully Connected Layer

FULLY

CONNECTED

LAYER

POOLING

LAYER

RECTIFIED

LINEAR

UNIT

CONVOLUTION

LAYER

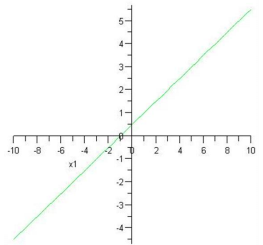
1. Convolution Layer

It gives the algorithm its distinctive name and also is used to extract features from the input image. Using a filter, the original matrix is modified normally into a smaller matrix. Although some of the features may be lost in the process, the main and unique features needed are preserved. The output array is usually referred to as Feature Map. To prevent the loss of important information, many feature maps can be created with different filters. The feature map of each filter has the ability to extract the position of specific features present in the input image. This layered approach has limitations that can be overcome by adding shortcut connections between layers as proposed in ResNet.



2.ReLu (Rectified Linear Unit) Layer

This operation takes all negative valued pixels in the feature map and replaces them with zero. Convolution is a linear process and therefore, we could commit the error of processing images as linear problems – which they are not. ReLu is simply used to input non-linearity back to the process. It is a linear graph.



3. Pooling Layer

The feature maps are scanned, by a fixed set of pixels, and the largest value from each window scanned is taken. This operation reduces the dimension of the feature maps making not only computation easier but also making the algorithm more robust to small perturbations and changes in the image

Max Pooling

It is the max element from every quadrant

|  |  |
| --- | --- |
| 43 | 112 |
| 45 | 56 |

Avg Pooling

It is the avg of all element from every quadrant

|  |  |
| --- | --- |
| 23 | 40 |
| 33 | 30 |

|  |  |  |  |
| --- | --- | --- | --- |
| 4 | 5 | 6 | 12 |
| 43 | 41 | 112 | 31 |
| 45 | 32 | 56 | 53 |
| 22 | 33 | 12 | 1 |

4. Full Connected Layer

The final layer in the CNN is the Fully Connected Layer (FCP). The objective of a fully connected layer is to take the results of the convolution/pooling process and use them to classify the image into a label (in a simple classification example). The output of convolution/pooling is flattened into a single vector of values, each representing a probability that a certain feature belongs to a label. For example, if the image is of a cat, features representing things like whiskers or fur should have high probabilities for the label “cat”.

|  |  |
| --- | --- |
| CAT | 75% |
| DOG | 15% |
| TURTLE | 10% |

* PRACTICE QUESTIONS

Q1. Find using Kernel.

|  |  |  |
| --- | --- | --- |
| -1 | 1 | 2 |
| 1 | 5 | 1 |
| 1 | 0 | 1 |

|  |  |  |
| --- | --- | --- |
| 2 | 2 | 2 |
| 2 | 3 | 3 |
| 1 | 1 | 1 |

Q2. Find max and average pooling for the following.

|  |  |  |  |
| --- | --- | --- | --- |
| 32 | 34 | 13 | 12 |
| 345 | 124 | 349 | 33 |
| 33 | 331 | 4 | 45 |
| 98 | 4 | 199 | 100 |