FACIAL EXPRESSION RECCOGNITION

A PROJECT REPORT

Submitted by

SACHIN SARKAR

UNIVERSITY REG NO. : 203002484310001

ARPAN DATTA

UNIVERSITY REG NO. : 203002484310003

ANIRBAN DEBNATH

UNIVERSITY REG NO. : 203002484310010

SAYAN MONDAL

UNIVERSITY REG NO. : 203002484310012

in partial fulfillment for the award of the degree of

BACHELOR OF SCIENCE

IN

INFORMATION TECHNOLOGY (DATA SCIENCE)

1ST YEAR, 2ND SEMESTER, YEAR: 2021

DEPARTMENT OF INFORMATION TECHNOLOGY

MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY

WEST BENGAL

BONAFIDE CERTIFICATE

Certified that this project report **FACIAL EXPRESSION RECCOGNITION** is the bonafide work of **SACHIN SARKAR, ARPAN DATTA, ANIRBAN DEBNATH, SAYAN MONDAL** who carried out the project work under my supervision.

MRS. SOMDATTA CHKRABORTY
HEAD OF THE DEPARTMENT
DEPARTMENT OF INFORMATION TECHNOLOGY
MAKAUT, WB

MRS. DIPIKA DHARA
SUPERVISOR
ASSISTANT PROFESSOR
DEPARTMENT OF IT, MAKAUT, WB

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to Mrs. Dipika Dhara, Assistant Professor of the department of Information Technology, whose role as project guide was invaluable for the project. We are extremely thankful for the keen interest he took in advising us, for the materials provided for the moral support extended to us.

Last but not the least we convey our gratitude to all the teachers for providing us the technical skill that will always remain as our asset and to all nonteaching staff for the gracious hospitality they offered us.

Place: MAKAUT, WB

Date: 7 AUG, 2021

Sachin Sarkar

Arpan Datta

Anirban Debnath

Sayan Mondal

Department of Information Technology

MAKAUT,HARINGHATA

NADIA-721249

West Bengal, India

ABSTRACT

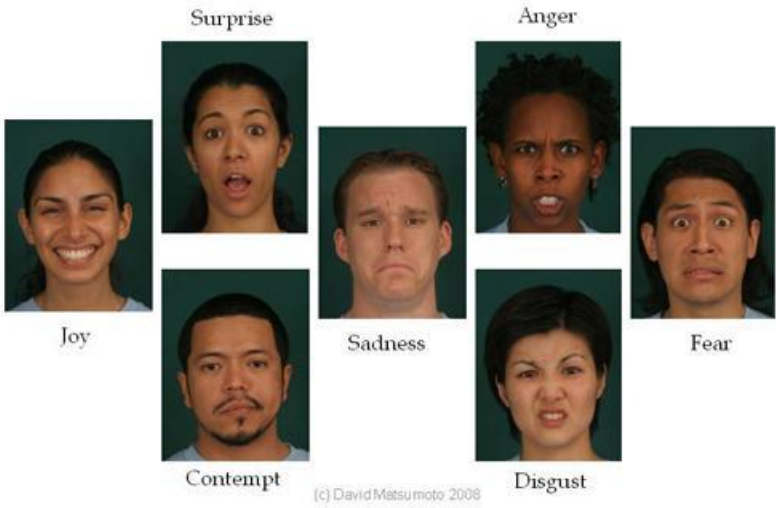
Facial expressions are the fastest means of communication while conveying any type of information. These are not only exposes the sensitivity or feelings of any person but can also be used to judge his/her mental views.
This project includes the introduction of the facial expression recognition and an investigation on the recent previous researches for extracting the effective and efficient method for facial expression recognition.
In this project we applied various deep learning methods (convolutional neural networks) to identify the key seven human emotions: anger, disgust, fear, happiness, sadness, surprise and neutrality.

TABLE OF CONTENTS

INTRODUCTION

With the advent of modern technology our desires went high and it binds no bounds. In the present era a huge research work is going on in the field of digital image and image processing. The way of progression has been exponential and it is ever increasing. Image Processing is a vast area of research in present day world and its applications are very widespread.

Image processing is the field of signal processing where both the input and output signals are images. One of the most important application of Image processing is Facial expression recognition. Our emotion is revealed by the expressions in our face.



Facial Expressions plays an important role in interpersonal communication. Facial expression is a nonverbal scientific gesture which gets expressed in our face as per our emotions. Automatic recognition of facial expression plays an important role in artificial intelligence and robotics and thus it is a need of the generation.

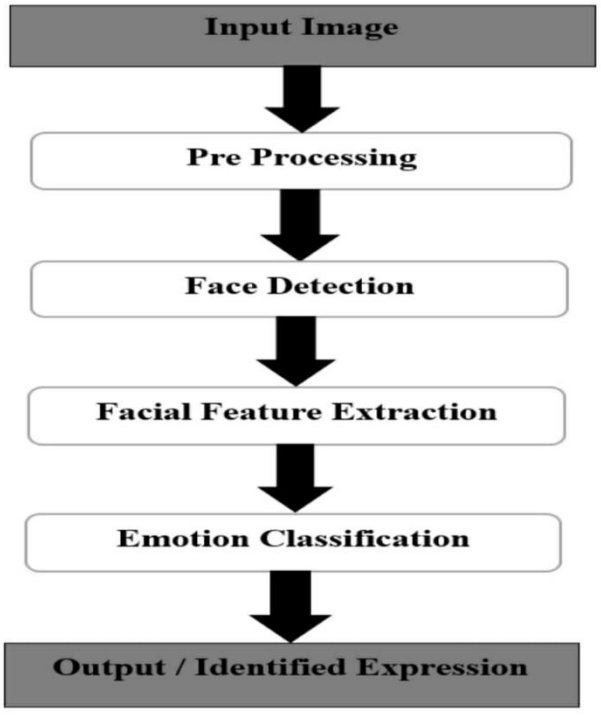
The objective of this project is to develop Automatic Facial Expression Recognition System which can take human facial images containing some expression as input and recognize and classify it into seven different expression class such as :(I. Angry II. Disgust III. Fear IV. Happy V. Neutral VI. Sadness VII. Surprise) Projects have already been done in this fields and our goal will not only be to develop an Automatic Facial Expression Recognition System but also improving the accuracy of this system compared to the other available systems.

PROBLEM DEFINATION

Human facial expressions can be easily classified into 7 basic emotions: happy, sad, surprise, fear, anger, disgust, and neutral. Our facial emotions are expressed through activation of specific sets of facial muscles. These sometimes subtle, yet complex, signals in an expression often contain an abundant amount of information about our state of mind. Through facial emotion recognition, we are able to measure the effects that content and services have on the audience/users through an easy and low-cost procedure. For example, retailers may use these metrics to evaluate customer interest. Healthcare providers can provide better service by using additional information about patients' emotional state during treatment. Entertainment producers can monitor audience engagement in events to consistently create desired content.

Humans are well-trained in reading the emotions of others, in fact, at just 14 months old, babies can already tell the difference between happy and sad. But can computers do a better job than us in accessing emotional states? To answer the question, We designed a deep learning neural network that gives machines the ability to make inferences about our emotional states. In other words, we give them eyes to see what we can see.

Problem formulation of our project:



Facial expression recognition is a process performed by humans or computers, which consists of:

1. Locating faces in the scene (e.g., in an image; this step is also referred to as facedetection),

2. Extracting facial features from the detected face region (e.g., detecting the shape of facialcomponents or describing the texture of the skin in a facial area; this step is referred to asfacial feature extraction),

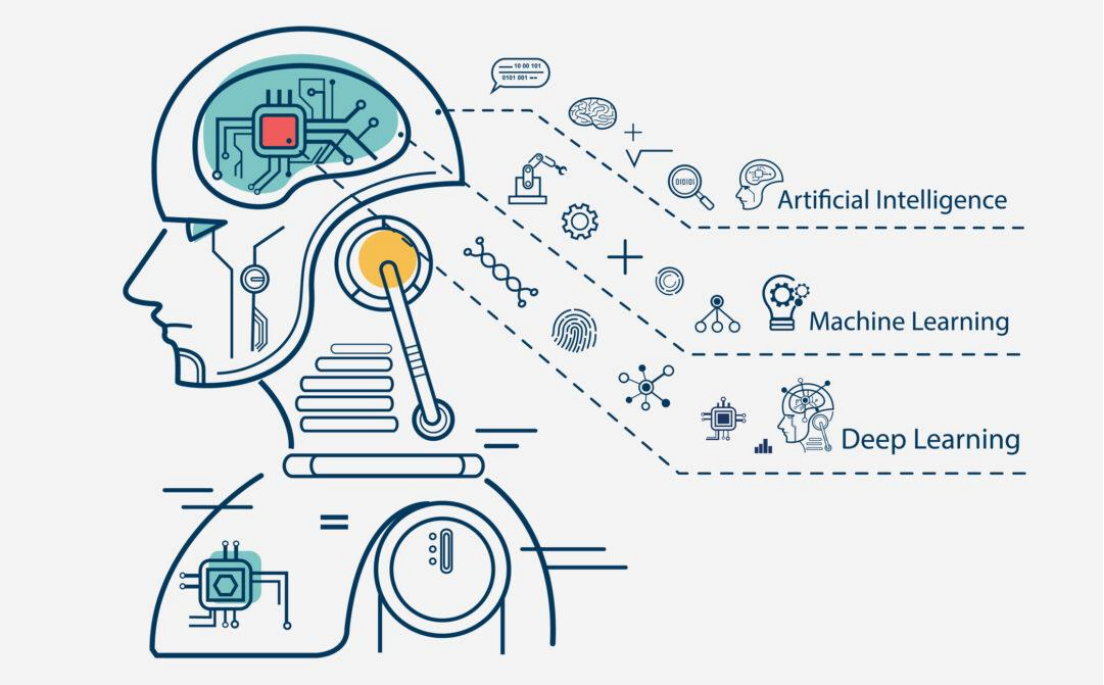
3. Analyzing the motion of facial features and/or the changes in the appearance of facial features and classifying this information into some facial-expression-interpretativecategories such as facial muscle activations like smile or frown, emotion (affect)categories like happiness or anger, attitude categories like (dis)liking or ambivalence, etc.(this step is also referred to as facial expression interpretation).

Several Projects have already been done in this fields and our goal will not only be to develop a Automatic Facial Expression Recognition System but also improving the accuracy of this system compared to the other available systems.

STUDY AREA AND RESEARCH

- **Deep Learning :**

Deep learning is a type of machine learning and artificial intelligence (AI) that imitates the way humans gain certain types of knowledge. Deep learning is an important element of data science, which includes statistics and predictive modeling. It is extremely beneficial to data scientists who are tasked with collecting, analyzing and interpreting large amounts of data; deep learning makes this process faster and easier.



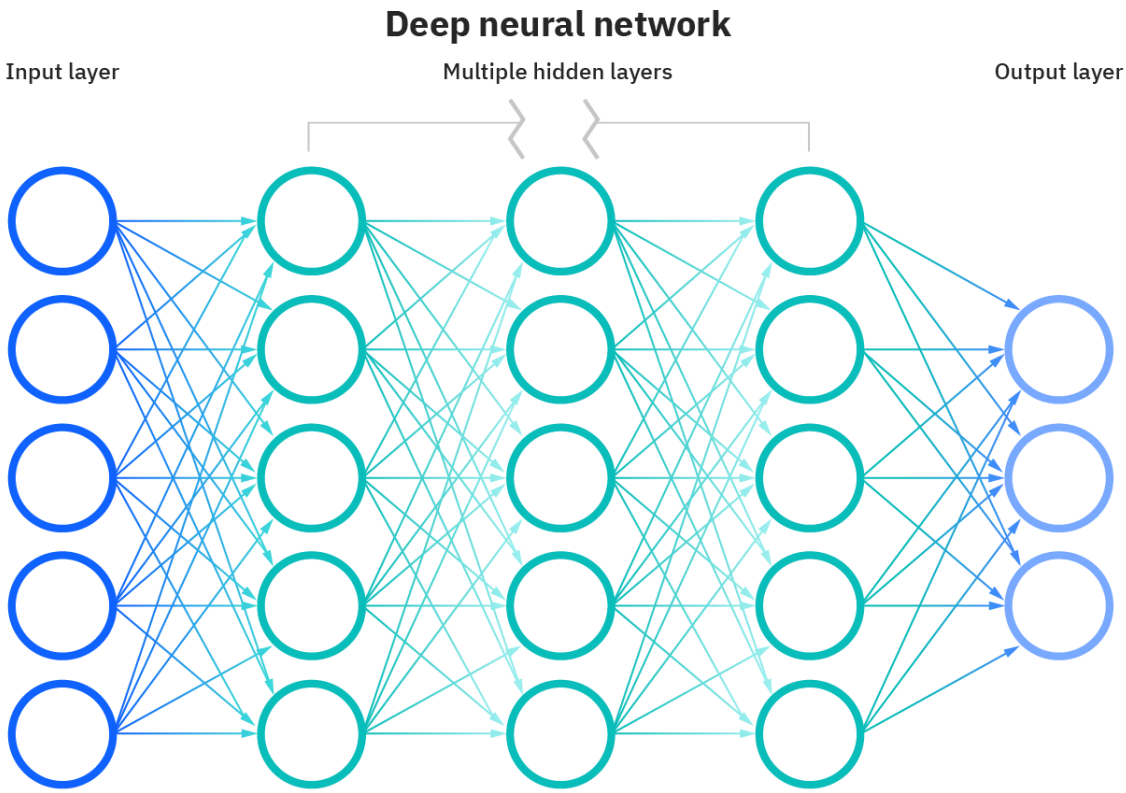
At its simplest, deep learning can be thought of as a way to automate predictive analytics. While traditional machine learning algorithms are linear, deep learning algorithms are stacked in a hierarchy of increasing complexity and abstraction.

To understand deep learning, imagine a toddler whose first word is dog. The toddler learns what a dog is -- and is not -- by pointing to objects and saying the word dog. The parent says, "Yes, that is a dog," or, "No, that is not a dog." As the toddler continues to point to objects, he

becomes more aware of the features that all dogs possess. What the toddler does, without knowing it, is clarify a complex abstraction -- the concept of dog -- by building a hierarchy in which each level of abstraction is created with knowledge that was gained from the preceding layer of the hierarchy.

- **Neural Networks :**

  Neural networks, also known as artificial neural networks (ANNs) or simulated neural networks (SNNs), are a subset of machine learning and are at the heart of deep learning algorithms. Their name and structure are inspired by the human brain, mimicking the way that biological neurons signal to one another.



Artificial neural networks (ANNs) are comprised of a node layers, containing an input layer, one or more hidden layers, and an output layer. Each node, or artificial neuron, connects to another and has an associated weight and threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network.

Neural networks rely on training data to learn and improve their accuracy over time. However, once these learning algorithms are fine-tuned for accuracy, they are powerful tools in computer science and artificial intelligence, allowing us to classify and cluster data at a high velocity. Tasks in speech recognition or image recognition can take minutes versus hours when compared to the manual identification by human experts. One of the most well-known neural networks is Google's search algorithm.

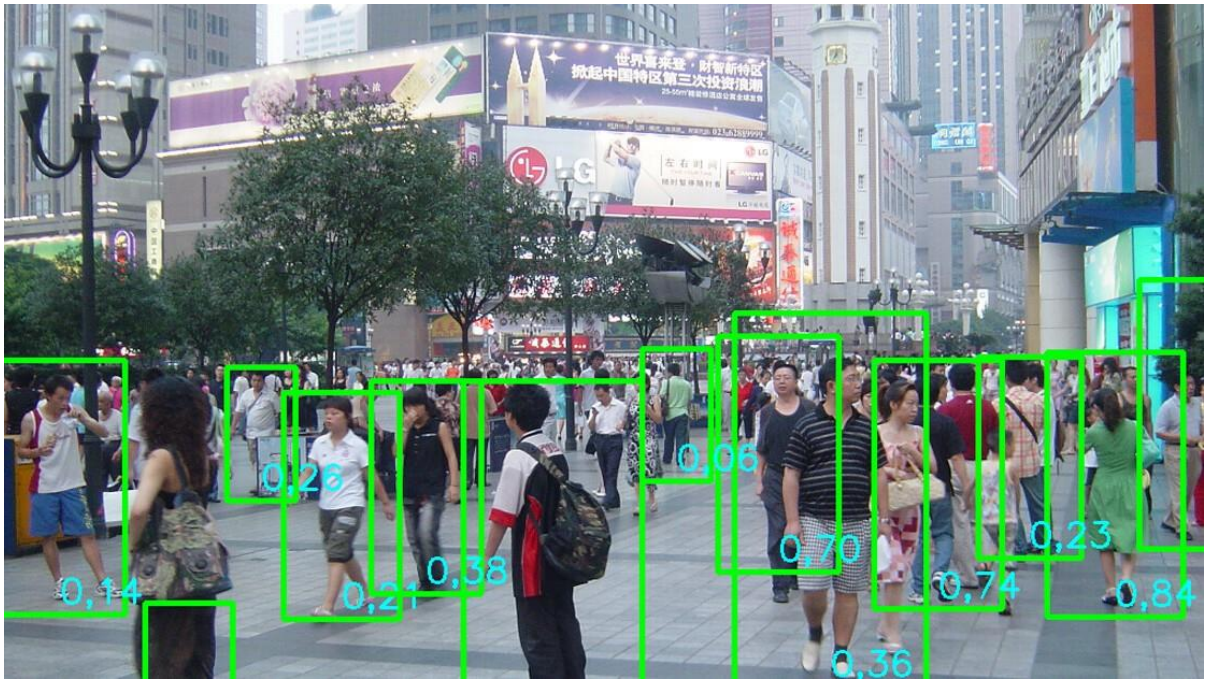
- **Computer Vision :**

Computer vision is a field of artificial intelligence (AI) that enables computers and systems to derive meaningful information from digital images, videos and other visual inputs — and take actions or make recommendations based on that information. If AI enables computers to think, computer vision enables them to see, observe and understand.

Computer vision works much the same as human vision, except humans have a head start. Human sight has the advantage of lifetimes of context to train how to tell objects apart, how far away they are, whether they are moving and whether there is something wrong in an image.

Computer vision trains machines to perform these functions, but it has to do it in much less time with cameras, data and algorithms rather than retinas, optic nerves and a visual cortex. Because a system trained to inspect products or watch a production asset can analyze thousands of products or processes a minute, noticing imperceptible defects or issues, it can quickly surpass human capabilities.

Computer vision is used in industries ranging from energy and utilities to manufacturing and automotive – and the market is continuing to grow. It is expected to reach USD 48.6 billion by 2022.



- **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.

There are two types of methods used for image processing namely, analogue and digital image processing. Analogue image processing can be used for the hard copies like printouts and photographs. Image analysts use various fundamentals of interpretation while using these visual techniques. Digital image processing techniques help in manipulation of the digital images by using computers. The three general phases that all types of data have to undergo while using digital technique are pre-processing, enhancement, and display, information extraction.

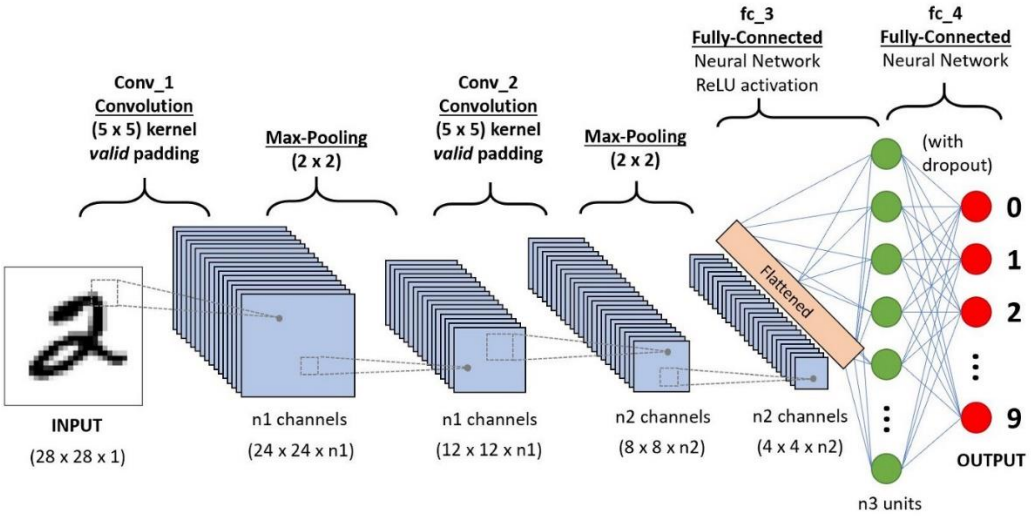
In this lecture we will talk about a few fundamental definitions such as image, digital image, and digital image processing. Different sources of digital images will be discussed and examples for each source will be provided. The continuum from image processing to computer vision will be covered in this lecture. Finally we will talk about image acquisition and different types of image sensors.

- # Convolutional Neural Network :

  - ## Introduction :

    A Convolutional Neural Network (ConvNet/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 pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

    The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area.



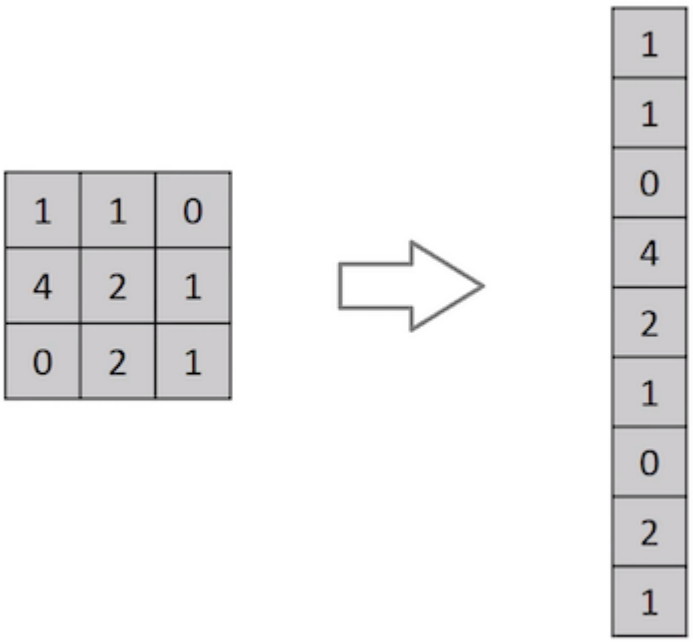
A CNN sequence to classify handwritten digits

- **Why ConvNets over Feed-Forward Neural Nets?**

  Flattening of a 3x3 image matrix into a 9x1 vector. An image is nothing but a matrix of pixel values, right? So why not just flatten the image (e.g. 3x3 image matrix into a 9x1 vector) and feed it to a Multi-Level Perceptron for classification purposes? Uh.. not really.

  In cases of extremely basic binary images, the method might show an average precision score while performing prediction of classes but would have little to no accuracy when it comes to complex images having pixel dependencies throughout.

  A ConvNet is able to successfully capture the Spatial and Temporal dependencies in an image through the application of relevant filters. The architecture performs a better fitting to the image dataset due to the reduction in the number of parameters involved and reusability of weights. In other words, the network can be trained to understand the sophistication of the image better.

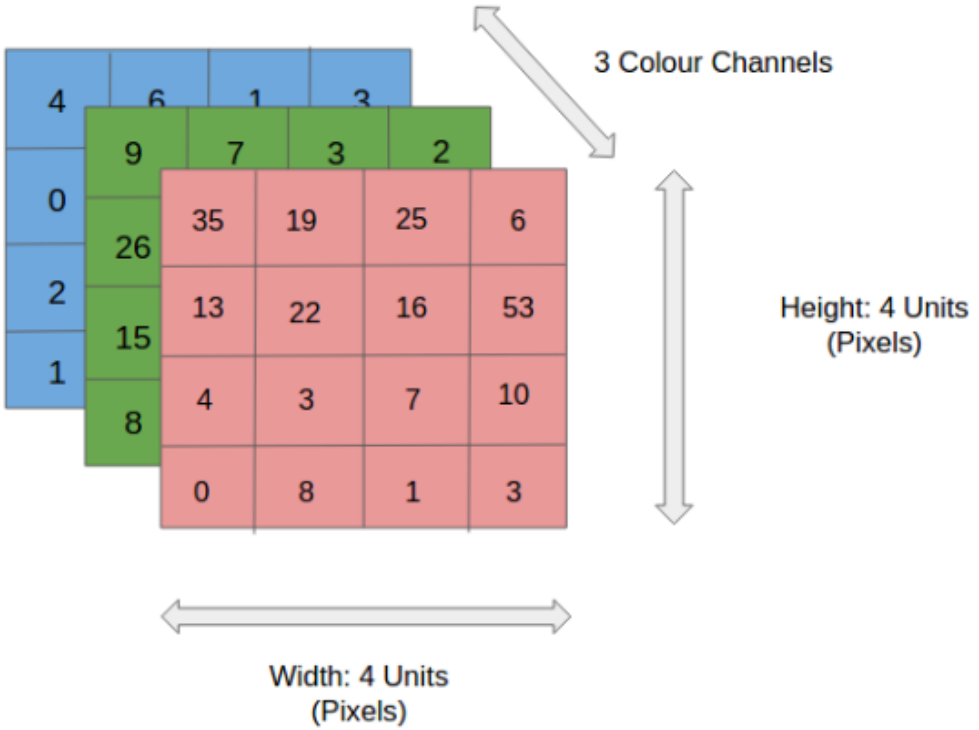


Flattening of a 3x3 image matrix into a 9x1 vector

- **Input Image :**

  In the figure, we have an RGB image which has been separated by its three color planes — Red, Green, and Blue. There are a number of such color spaces in which images exist — Grayscale, RGB, HSV, CMYK, etc.

  You can imagine how computationally intensive things would get once the images reach dimensions, say 8K (7680×4320). The role of the ConvNet is to reduce the images into a form which is easier to process, without losing features which are critical for getting a good prediction. This is important when we are to design an architecture which is not only good at learning features but also is scalable to massive datasets.

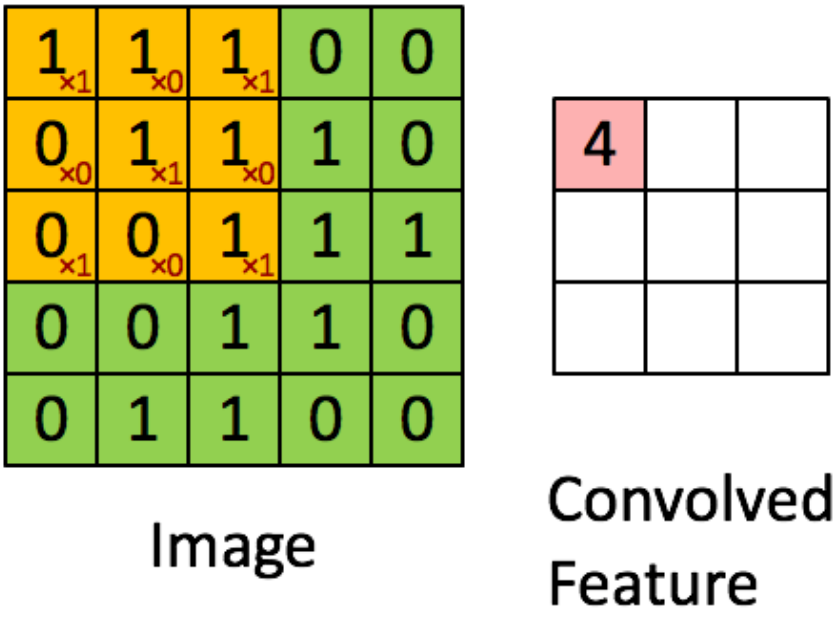


4x4x3 RGB Image

- **Convolution Layer — The Kernel :**

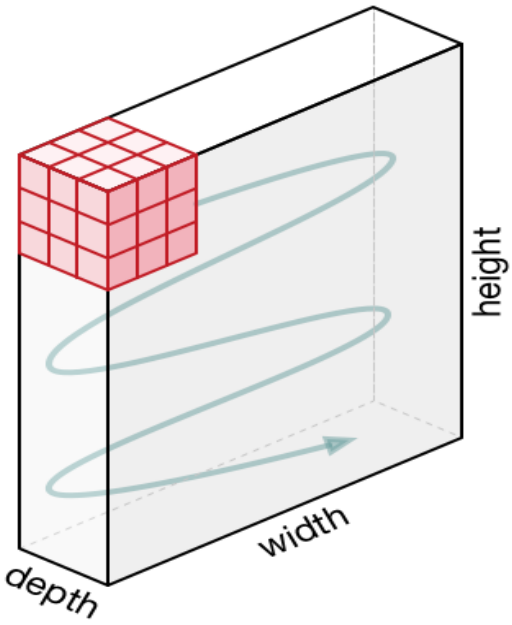
  Image Dimensions = 5 (Height) x 5 (Breadth) x 1 (Number of channels, eg. RGB)

  In the above demonstration, the green section resembles our 5x5x1 input image, I. The element involved in carrying out the convolution operation in the first part of a Convolutional Layer is called the Kernel/Filter, K, represented in the color yellow. We have selected K as a 3x3x1 matrix.



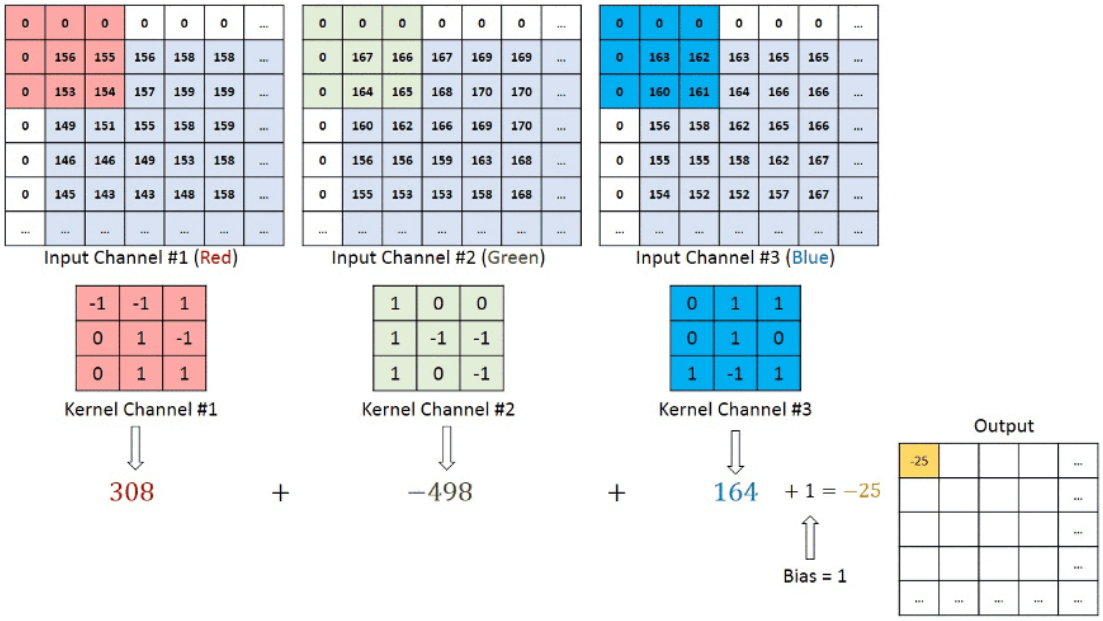
Convoluting a 5x5x1 image with a 3x3x1 kernel to get a 3x3x1 convolved feature

The Kernel shifts 9 times because of Stride Length = 1 (Non-Strided), every time performing a matrix multiplication operation between K and the portion P of the image over which the kernel is hovering.



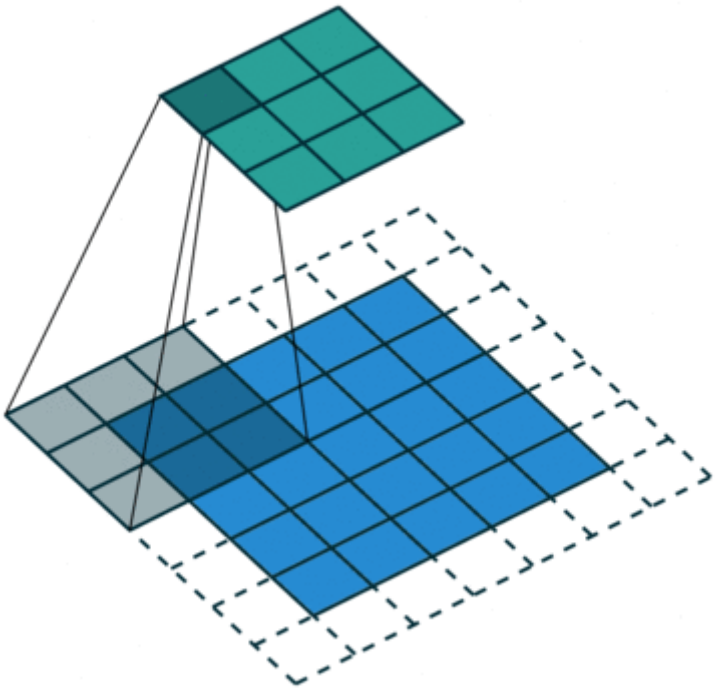
Movement of the Kernel

The filter moves to the right with a certain Stride Value till it parses the complete width. Moving on, it hops down to the beginning (left) of the image with the same Stride Value and repeats the process until the entire image is traversed.



Convolution operation on a MxNx3 image matrix with a 3x3x3 Kernel

In the case of images with multiple channels (e.g. RGB), the Kernel has the same depth as that of the input image. Matrix Multiplication is performed between Kn and In stack ([K1, I1]; [K2, I2]; [K3, I3]) and all the results are summed with the bias to give us a squashed one-depth channel Convoluted Feature Output.



Convolution Operation with Stride Length = 2

The objective of the Convolution Operation is to extract the high-level features such as edges, from the input image. ConvNets need not be limited to only one Convolutional Layer. Conventionally, the first ConvLayer is responsible for capturing the Low-Level features such as edges, color, gradient orientation, etc. With added layers, the architecture adapts to the High-Level features as well, giving us a network which has the wholesome understanding of images in the dataset, similar to how we would.

There are two types of results to the operation — one in which the convolved feature is reduced in dimensionality as compared to the input, and the other in which the dimensionality is either increased or remains the same. This is done by applying Valid Padding in case of the former, or Same Padding in the case of the latter.

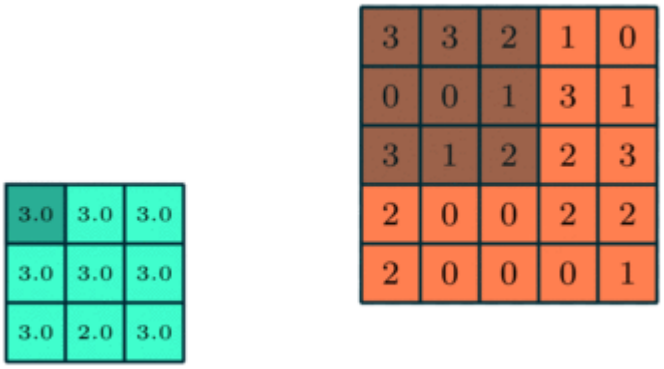
When we augment the 5x5x1 image into a 6x6x1 image and then apply the 3x3x1 kernel over it, we find that the convolved matrix turns out to be of dimensions 5x5x1. Hence the name — Same Padding.

On the other hand, if we perform the same operation without padding, we are presented with a matrix which has dimensions of the Kernel (3x3x1) itself — Valid Padding.

The following repository houses many such GIFs which would help you get a better understanding of how Padding and Stride Length work together to achieve results relevant to our needs.

- **Pooling Layer :**

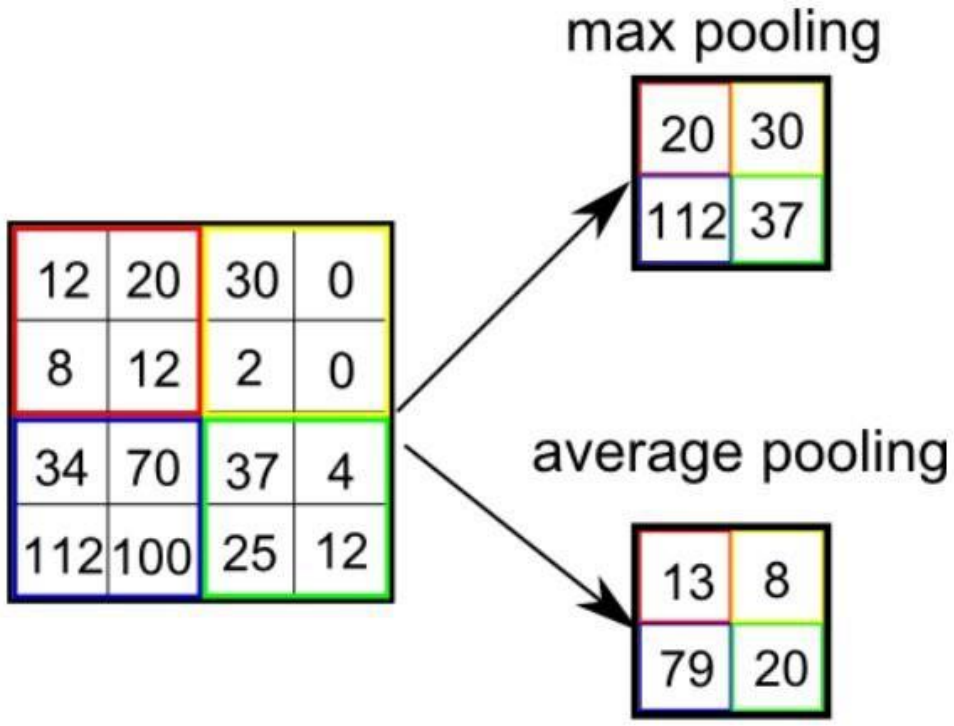
Similar to the Convolutional Layer, the Pooling layer is responsible for reducing the spatial size of the Convolved Feature. This is to decrease the computational power required to process the data through dimensionality reduction. Furthermore, it is useful for extracting dominant features which are rotational and positional invariant, thus maintaining the process of effectively training of the model.



3x3 pooling over 5x5 convolved feature

There are two types of Pooling: Max Pooling and Average Pooling. Max Pooling returns the maximum value from the portion of the image covered by the Kernel. On the other hand, Average Pooling returns the average of all the values from the portion of the image covered by the Kernel.

Max Pooling also performs as a Noise Suppressant. It discards the noisy activations altogether and also performs de-noising along with dimensionality reduction. On the other hand, Average Pooling simply performs dimensionality reduction as a noise suppressing mechanism. Hence, we can say that Max Pooling performs a lot better than Average Pooling.

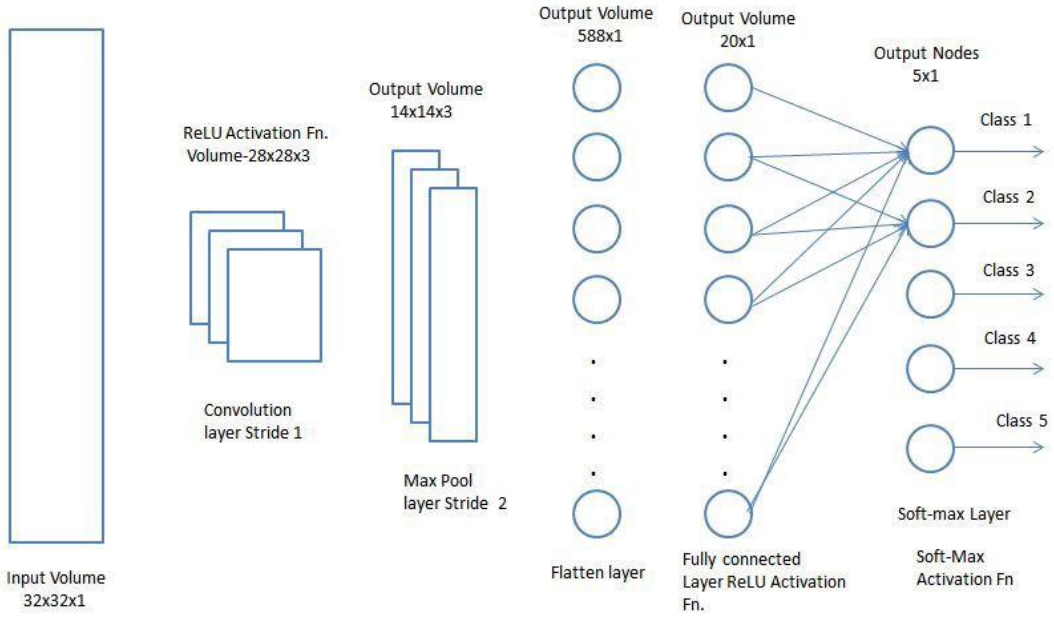


Types of Pooling

The Convolutional Layer and the Pooling Layer, together form the i-th layer of a Convolutional Neural Network. Depending on the complexities in the images, the number of such layers may be increased for capturing low-levels details even further, but at the cost of more computational power.

After going through the above process, we have successfully enabled the model to understand the features. Moving on, we are going to flatten the final output and feed it to a regular Neural Network for classification purposes.

- **Classification — Fully Connected Layer (FC Layer) :**



Adding a Fully-Connected layer is a (usually) cheap way of learning non-linear combinations of the high-level features as represented by the output of the convolutional layer. The Fully-Connected layer is learning a possibly non-linear function in that space.

Now that we have converted our input image into a suitable form for our Multi-Level Perceptron, we shall flatten the image into a column vector. The flattened output is fed to a feed-forward neural network and backpropagation applied to every iteration of training. Over a series of epochs, the model is able to distinguish between dominating and certain low-level features in images and classify them using the Softmax Classification technique.

There are various architectures of CNNs available which have been key in building algorithms which power and shall power AI as a whole in the foreseeable future. Some of them have been listed below:

- LeNet
- AlexNet
- VGGNet
- GoogLeNet
- ResNet
- ZFNet

PROJECT IMPLEMENTATION AND METHODOLOGY

- **About the Dataset :**

The dataset, used for training the model is from a Kaggle Facial Expression ecognition Challenge a few years back (FER2013). The data consists of 48x48 pixel grayscale images of faces. The faces have been automatically registered so that the face is more or less centered and occupies about the same amount of space in each image. The task is to categorize each face based on the emotion shown in the facial expression in to one of seven categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral).

The training set consists of 28,709 examples. The public test set used for the leaderboard consists of 3,589 examples. The final test set, which was used to determine the winner of the competition, consists of another 3,589 examples. Emotion labels in the dataset:

0: -4593 images- Angry

1: -547 images- Disgust

2: -5121 images- Fear

3: -8989 images- Happy

4: -6077 images- Sad

5: -4002 images- Surprise

6: -6198 images- Neutral

```
[11] dataset = pd.read_csv('/content/drive/MyDrive/face_emotions/fer2013.csv')
     print(dataset.head())
     print(dataset.info())
```

```
   emotion                                             pixels     Usage
0        0  70 80 82 72 58 58 60 63 54 58 60 48 89 115 121...  Training
1        0  151 150 147 155 148 133 111 140 170 174 182 15...  Training
2        2  231 212 156 164 174 138 161 173 182 200 106 38...  Training
3        4  24 32 36 30 32 23 19 20 30 41 21 22 32 34 21 1...  Training
4        6  4 0 0 0 0 0 0 0 0 0 0 0 3 15 23 28 48 50 58 84...  Training
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 35887 entries, 0 to 35886
Data columns (total 3 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   emotion  35887 non-null  int64
 1   pixels   35887 non-null  object
 2   Usage    35887 non-null  object
dtypes: int64(1), object(2)
memory usage: 841.2+ KB
None
```

```
data.head()
```

	emotion	pixels	Usage
0	0	70 80 82 72 58 58 60 63 54 58 60 48 89 115 121...	Training
1	0	151 150 147 155 148 133 111 140 170 174 182 15...	Training
2	2	231 212 156 164 174 138 161 173 182 200 106 38...	Training
3	4	24 32 36 30 32 23 19 20 30 41 21 22 32 34 21 1...	Training
4	6	4 0 0 0 0 0 0 0 0 0 0 0 3 15 23 28 48 50 58 84...	Training

FER2013 dataset in csv form

- **Images Preprocessing :**

In this section we preprocessing our dataset. The data divided into 4 python list x_train, y_train, x_test, y_test as per need for training and validate the Model.

We divide the dataset for training and validating on the basis of Usage column such as we filter out all training dataset by ‘Training’ in Usage column and ‘PublicTest’ in Usage column.

A total of 28709 images listed in x_train for training purpose and a total of 3589 images listed in x_test to validate the model.

After that we store the list into numpy array with data type ‘float32’.

now rescale the all pixel values into range of 0-1.

All the images reshaped into 48*48*1 nd array.

Now our training and testing data are ready to be trained successfully in the next step.

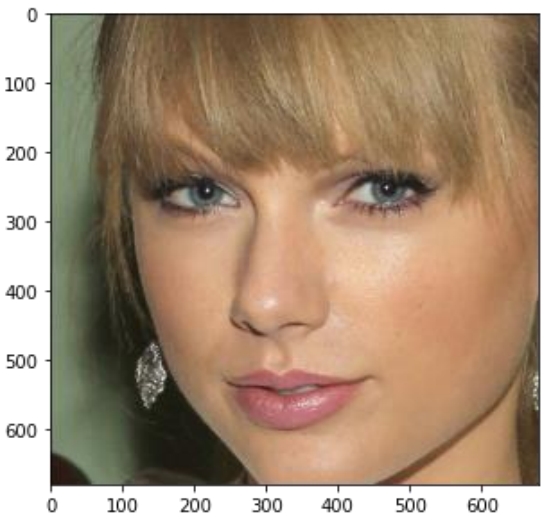


Image before Preprocessing

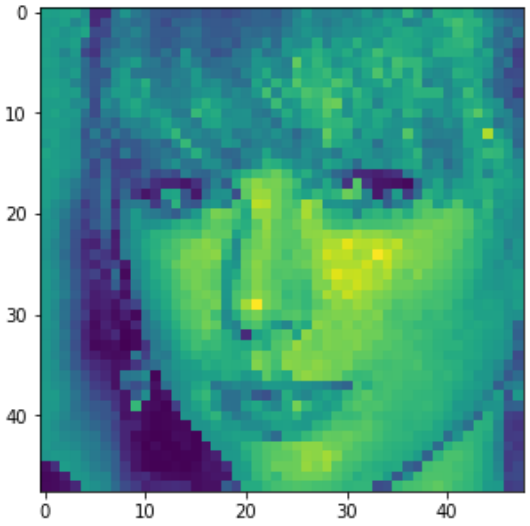


Image after Preprocessing

Let divide the Dataset into Training and Testing

```
[4]  x_train,y_train,x_test,y_test = [],[],[],[]
     for index,row in dataset.iterrows():
       val = row['pixels'].split(' ')
       try:
         if 'Training' in row['Usage']:
           x_train.append(np.array(val,'float32'))
           y_train.append(row['emotion'])
         elif 'PublicTest' in row['Usage']:
           x_test.append(np.array(val,'float32'))
           y_test.append(row['emotion'])
       except:
         print("Error occured at index :",index,"and row :",row)
```

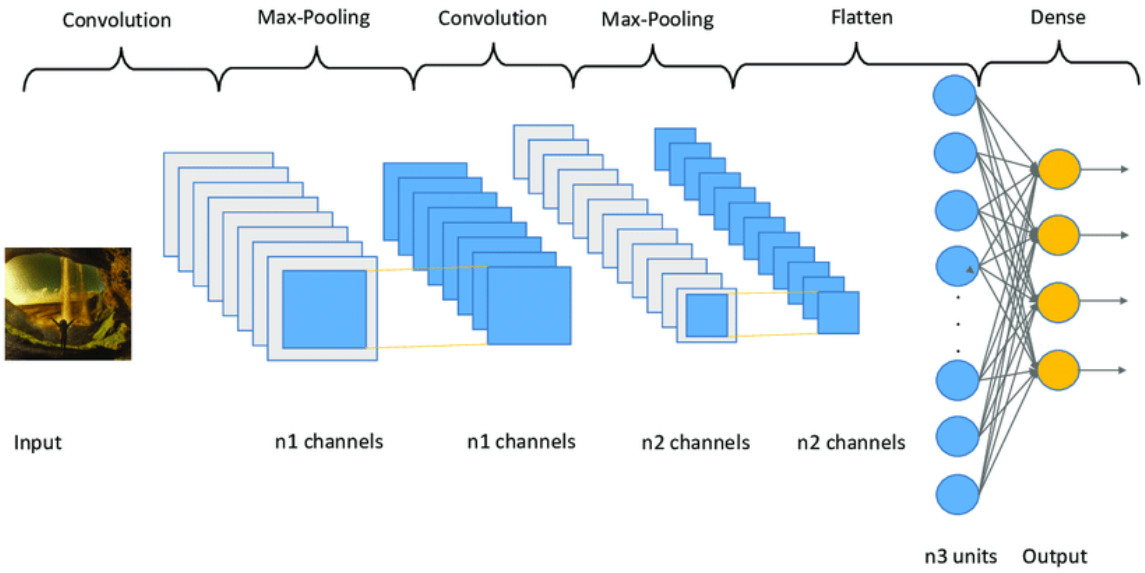
Preprocessing the Dataset

```
x_train = np.array(x_train,'float32')
y_train = np.array(y_train,'float32')
x_test = np.array(x_test,'float32')
y_test = np.array(y_test,'float32')

y_train = to_categorical(y_train,num_classes=7)
y_test = to_categorical(y_test,num_classes=7)

x_train /= 255.0
x_test /= 255.0

x_train = x_train.reshape(x_train.shape[0],48,48,1)
x_test = x_test.reshape(x_test.shape[0],48,48,1)

print("x_train Shape : ",x_train.shape)
print("x_test.shape : ",x_test.shape)
```

```
x_train Shape :  (28709, 48, 48, 1)
x_test.shape :  (3589, 48, 48, 1)
```

Code for Pre-processing

- **Covolutional Neural Network Model :**



Convolutional Neural network Architechture

Here we create our own CNN Model to Extact the features from the Images in order to so that our Model will able to differ the different different images associated with different different emotions.

In this CNN model we added the total of 8 convolutional Layers, with the same kernel size of (3,3) associated with 8 Relu layers.
As same as Convolutional Layers we added many layers such as we added MaxPooling as Pooling layers, use Batch Normalization and some Dropout layers also so that our model will deeper but not over-fitted as well.

Then we Flatten the data and added fully connected layer Dense with activation layers Relu, Softmax at the end to classify the images by reading pixel values it contains.

CNN Model Building

```
num_features = 64
num_labels = 7

model = Sequential()

model.add(Conv2D(num_features, kernel_size=(3, 3), activation='relu', input_shape=(48,48,1),
                 data_format='channels_last', kernel_regularizer=l2(0.01)))
model.add(Conv2D(num_features, kernel_size=(3, 3), activation='relu', padding='same'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
model.add(Dropout(0.5))

model.add(Conv2D(2*num_features, kernel_size=(3, 3), activation='relu', padding='same'))
model.add(BatchNormalization())
model.add(Conv2D(2*num_features, kernel_size=(3, 3), activation='relu', padding='same'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
model.add(Dropout(0.5))
```

```
model.add(Conv2D(2*2*num_features, kernel_size=(3, 3), activation='relu', padding='same'))
model.add(BatchNormalization())
model.add(Conv2D(2*2*num_features, kernel_size=(3, 3), activation='relu', padding='same'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
model.add(Dropout(0.5))

model.add(Conv2D(2*2*2*num_features, kernel_size=(3, 3), activation='relu', padding='same'))
model.add(BatchNormalization())
model.add(Conv2D(2*2*2*num_features, kernel_size=(3, 3), activation='relu', padding='same'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
model.add(Dropout(0.5))

model.add(Flatten())
```

```
model.add(Dense(2*2*2*num_features, activation='relu'))
model.add(Dropout(0.4))
model.add(Dense(2*2*num_features, activation='relu'))
model.add(Dropout(0.4))
model.add(Dense(2*num_features, activation='relu'))
model.add(Dropout(0.5))

model.add(Dense(num_labels, activation='softmax'))

model.summary()
```

CNN Model Implementation

- **Compile and Train Model :**

  As our model is ready to be trained by FER2013 Dataset, we need to compile and fit the data to model so that our model will predict or classify easily.
  To compile our model we used ‘adam’ as an optimizer, ‘categorical_crossentropy’ as loss detector and metrics ‘accuracy’ to check the accuracy per epoch.
  Now to train our model here we used model.fit() method with 100 number of epochs and 287 no. of image arrays per batch.
  Let have a look to the accuracy score :

Compile and Fitting training dataset to the CNN Model

```
[7]  model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
     model.fit(x_train,y_train,batch_size=287,epochs=100,verbose=1,validation_data=(x_test,y_test),shuffle=True)
```

```
101/101 [==============================] - 18s 178ms/step - loss: 0.3936 - accuracy: 0.8723 - val_loss: 1.4345 - val_accuracy: 0.6342
Epoch 91/100
101/101 [==============================] - 18s 178ms/step - loss: 0.3538 - accuracy: 0.8844 - val_loss: 1.2815 - val_accuracy: 0.6464
Epoch 92/100
101/101 [==============================] - 18s 178ms/step - loss: 0.3392 - accuracy: 0.8878 - val_loss: 1.4429 - val_accuracy: 0.6347
Epoch 93/100
101/101 [==============================] - 18s 178ms/step - loss: 0.3867 - accuracy: 0.8748 - val_loss: 1.4861 - val_accuracy: 0.6453
Epoch 94/100
101/101 [==============================] - 18s 178ms/step - loss: 0.3411 - accuracy: 0.8917 - val_loss: 1.4176 - val_accuracy: 0.6503
Epoch 95/100
101/101 [==============================] - 18s 178ms/step - loss: 0.3451 - accuracy: 0.8880 - val_loss: 1.4580 - val_accuracy: 0.6294
Epoch 96/100
101/101 [==============================] - 18s 179ms/step - loss: 0.3926 - accuracy: 0.8726 - val_loss: 1.4119 - val_accuracy: 0.6459
Epoch 97/100
101/101 [==============================] - 18s 179ms/step - loss: 0.3454 - accuracy: 0.8877 - val_loss: 1.4002 - val_accuracy: 0.6500
Epoch 98/100
101/101 [==============================] - 18s 179ms/step - loss: 0.3825 - accuracy: 0.8783 - val_loss: 1.3386 - val_accuracy: 0.6553
Epoch 99/100
101/101 [==============================] - 18s 178ms/step - loss: 0.3198 - accuracy: 0.8964 - val_loss: 1.3899 - val_accuracy: 0.6620
Epoch 100/100
101/101 [==============================] - 18s 179ms/step - loss: 0.3011 - accuracy: 0.9021 - val_loss: 1.4143 - val_accuracy: 0.6601
<keras.callbacks.History at 0x7fd7061bdb90>
```

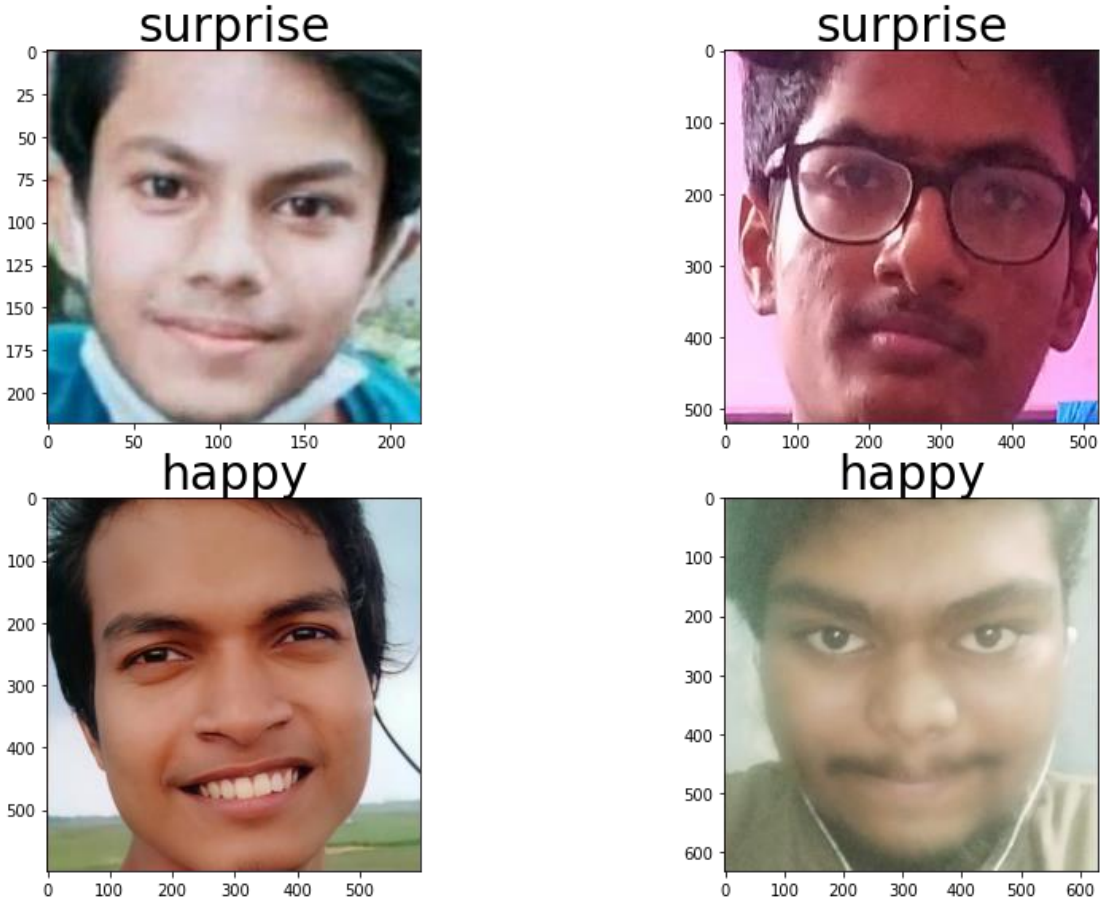
Compile and fit data to model

As per the screenshot of the accuracy, we can easily say that our trained successfully with 90.21% of accuracy on training data and 66.01% of accuracy on validation data which is much enough if we compare it to some other pre-build models.

- **Experiment and Result :**

  As per the previous steps we knows that our CNN Facial Experiment Recognition trained successfully with nearly 90-91% of accuracy on training dataset and 66% accuracy on testing dataset.

  Now its time to test or experiment our model. To experiment our model we take 4 images of ours.
  Now before we feed this images to our model to predict we need to preprocessed this all as per the model requirments.
  After preprocessing now we feed this image and we get the output that showed below and from a brief discussion, we analyze and conclude that the model is well predict our emotions with an discussed accuracy of 70-75%.



Classified the Experimented Images

FUTURE SCOPE

It is important to note that there is no specific formula to build a neural network that would guarantee to work well. Different problems would require different network architecture and a lot of trail and errors to produce desirable validation accuracy. This is the reason why neural nets are often perceived as "black box algorithms.". In this project we got an accuracy of almost 60% which is not bad at all comparing all the previous models. But we need to improve in specific areas like-

- number and configuration of convolutional layers
- number and configuration of dense layers
- dropout percentage in dense layers

But due to lack of highly configured system we could not go deeper into dense neural network as the system gets very slow and we will try to improve in these areas in future.We would also like to train more databases into the system to make the model more and more accurate but again resources becomes a hindrance in the path and we also need to improve in several areas in future to resolve the errors and improve the accuracy. Having examined techniques to cope with expression variation, in future it may be investigated in more depth about the face classification problem and optimal fusion of color and depth information. Further study can be laid down in the direction of allele of gene matching to the geometric factors of the facial expressions. The genetic property evolution framework for facial expressional system can be studied to suit the requirement of different security models such as criminal detection, governmental confidential security breaches etc.

CONCLUSION

The facial expression recognition system presented in this research work contributes a resilient face recognition model based on the mapping of behavioral characteristics with the physiological biometric characteristics. The physiological characteristics of the human face with relevance to various expressions such as happiness, sadness, fear, anger, surprise and disgust are associated with geometrical structures which restored as base matching emplate for the recognition system.The behavioral aspect of this system relates the attitude behind different expressions as property base. The property bases are alienated as exposed and hidden category in genetic algorithmic genes. The gene training set evaluates the expressional uniqueness of individual faces and provide a resilient expressional recognition model in the field of biometric security. The design of a novel asymmetric cryptosystem based on biometrics having features like hierarchical group security eliminates the use of passwords and smart cards as opposed to earlier cryptosystems. It requires a special hardware support like all other biometrics system. This research work promises a new direction of research in the field of asymmetric biometric cryptosystems which is highly desirable in order to get rid of passwords and smart cards completely. Experimental analysis and study show that the hierarchical security structures are effective in geometric shape identification for physiological traits.

REFERENCES

 A literature survey on Facial Expression Recognition using Global Featuresby Vaibhavkumar J. Mistry and Mahesh M. Goyani,International Journal of Engineering and Advanced Technology (IJEAT),April,2013 [http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.645.5162&rep=rep1&type=pdf]

 Convolutional Neural Networks (CNN) With TensorFlow by Sourav from Edureka[https://www.youtube.com/watch?v=umGJ30-15_A]

 Recognizing Facial Expressions Using Deep Learning by Alexandru Savoiu Stanford University and James Wong Stanford University [http://cs231n.stanford.edu/reports/2017/pdfs/224.pdf]

 Deep Learning Simplified by Sourav from Edureka [https://www.youtube.com/watch?v=dafuAz_CV7Q&list=PL9ooVrP1hQOEX8BKDplfG86ky8s7Oxbzg]

 Predicting facial expressions with machine learning algorithms by Alex Young , Andreas Eliasson , Ara Hayrabedian , Lukas Weiss , Utku Ozbulak[https://github.com/utkuozbulak/facial-expression-recognition]

 “Robust Real-Time Face Detection”, International Journal of Computer Vision 57(2), 137–154, 2004

 “Facial expressions of emotions: an old controversy and new finding discussion”, by P. Ekman, E. T. Rolls, D. I. Perrett, H. D. Ellis, Pill Trans. Royal Soc. London Ser. B, Biol. Sci., 1992, vol. 335, no. 1273, pp. 63-69.

 Going Deeper in Facial Expression Recognition using Deep Neural Networks, by Ali Mollahosseini1, David Chan2, and Mohammad H. Mahoor1 Department of Electrical and Computer Engineering, Department of Computer Science, University of Denver, Denver, CO.