REPORT ON HUMAN EMOTION RECOGNITION

REPORT SUBMITTED BY:

EKTA GUPTA
ANJALI SILAWAT
DEEPANSHI CHAURASIA

TO: RITU JANGRA

INTRODUCTION

WHAT IS HUMAN EMOTION?

Emotions are mental states brought on by neurophysiological changes, variously associated with thoughts, feelings, behavioural responses, and a degree of pleasure or displeasure. There is currently no scientific consensus on a definition. Emotions are often intertwined with mood, temperament, personality, disposition, or creativity.

Research on emotion has increased over the past two decades with many fields contributing including psychology, medicine, history, sociology of emotions, and computer science. The numerous theories that attempt to explain the origin, function and other aspects of emotions have fostered more intense research on this topic. Current areas of research in the concept of emotion include the development of materials that stimulate and elicit emotion. In addition, PET scans and fMRI scans help study the affective picture processes in the brain.

It’s contribution to computer science

In the 2000s, research in computer science, engineering, psychology and neuroscience has been aimed at developing devices that recognize human affect display and model emotions. In computer science, affective computing is a branch of the study and development of artificial intelligence that deals with the design of systems and devices that can recognize, interpret, and process human emotions. It is an interdisciplinary field spanning computer sciences, psychology, and cognitive science. While the origins of the field may be traced as far back as to early philosophical enquiries into emotion, the more modern branch of computer science originated with Rosalind Picard's 1995 paper on affective computing. Detecting emotional information begins with passive sensors which capture data about the user's physical state or behavior without interpreting the input. The data gathered is analogous to the cues humans use to perceive emotions in others. Another area within affective computing is the design of computational devices proposed to exhibit either innate emotional capabilities or that are capable of convincingly simulating emotions. Emotional speech processing recognizes the user's emotional state by analyzing speech patterns. The detection and processing of facial expression or body gestures is achieved through detectors and sensors.

DIFFERENT TYPES OF FACIAL EMOTION

1.)HAPPY A pleasant emotional state that elicits feelings of joy, contentment and satisfaction
Expression: Smile, laughter

2.)SAD An emotional state characterized by feelings of disappointment, grief or hopelessness
Expression: Frown, loss of focus in eyes, tears

3.)CRYING

4.)NEUTRAL

5.)BORED

6.)CONFUSED

7.)SURPRISED A brief emotional state, either positive or negative, following something unexpected
Expression: Raised brows, open mouth, gasp

8.)ANGER An emotional state leading to feelings of hostility and frustration
Expression: Glare, eyebrows drawn together, tight lips

9.)DISGUST A strong emotion that results in feeling repulsed
Expression: Wrinkled nose, gagging, no eye contact

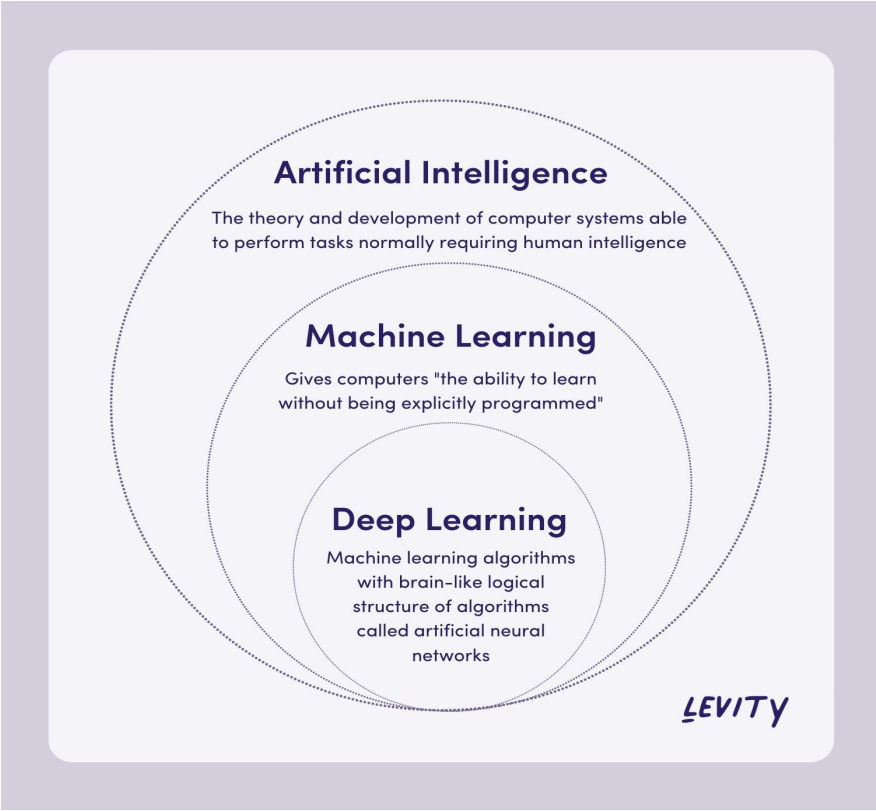
10.)FEAR A primal emotion that is important to survival and triggers a fight or flight response
Expression: Wide eyes, tense stretched lips

Concept used for detection of human emotion using AI

INTRODUCTION TO DEEP LEARNING

Deep Learning (also known as deep learning) is part of a broader family of machine learning methods based on artificial neural networks with representation learning. Learning can supervised, semi-supervised or unsupervised

Deep-learning architectures such as deep neural networks, deep belief networks, deep reinforcement learning, recurrent neural networks and convolutional neural networks have been applied to fields including computer vision, speech recognition, natural language processing, machine translation, bioinformatics, drug design, medical image analysis, climate science, material inspection and board game programs, where they have produced results comparable to and in some cases surpassing human expert performance



HUMAN NEURON

Cell Nucleus

Dendrites

Synapse

Axon

Terminal axon

NEURAL NETWORK

Node

Input: x1, x2

Weights or interconnections

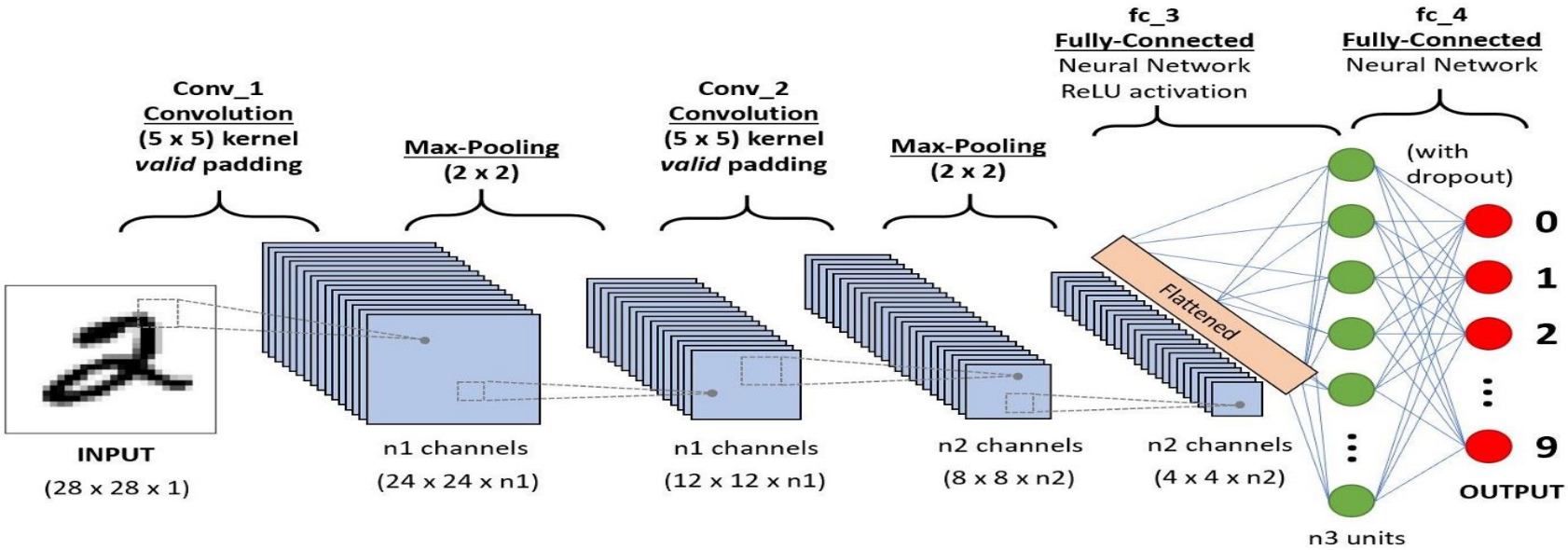
f(x) (mapping, activation, learning)

Output: y1

CONVOLUTIONAL NEURAL NETWORKS

Convolutional neural networks are a specialized type of artificial Neural networks that use a mathematical operation called convolution in place of general matrix multiplication in at least one of their layers. They are specifically designed

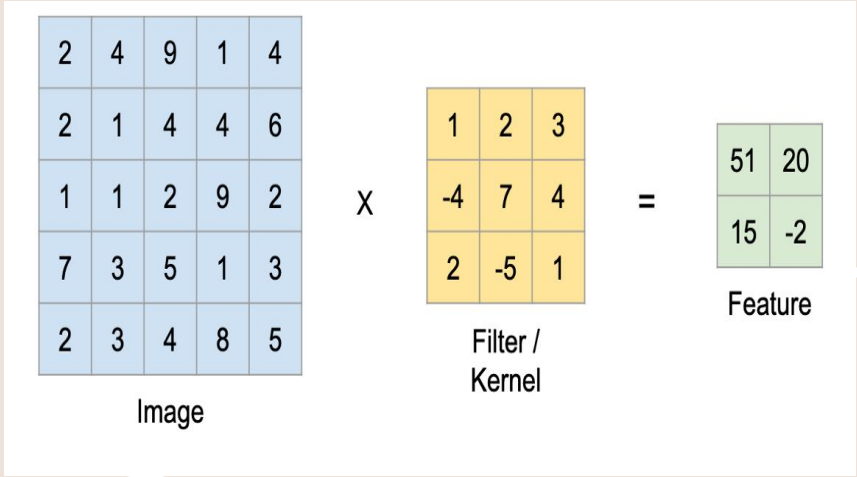
to process pixel data and are used in image recognition and processing.CNNs are also known as shift Invariant or Space Invariant Artificial Neural Networks(SIANN), based on the shared-weight architecture of the convolution kernels or filters that slide along input features and provide translation-equivariant responses known as features maps.



Layers in CNN

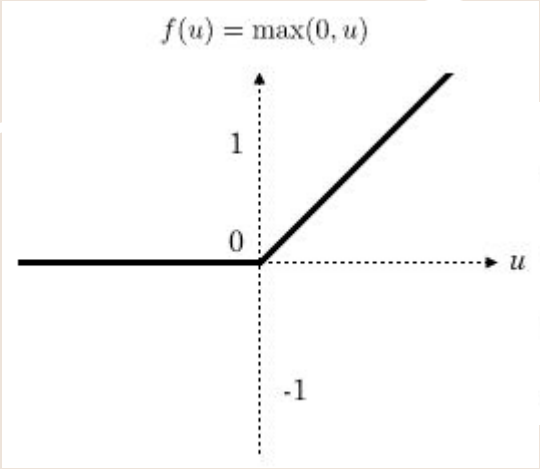
1.CONVOLUTIONAL LAYER "converts images into an array"

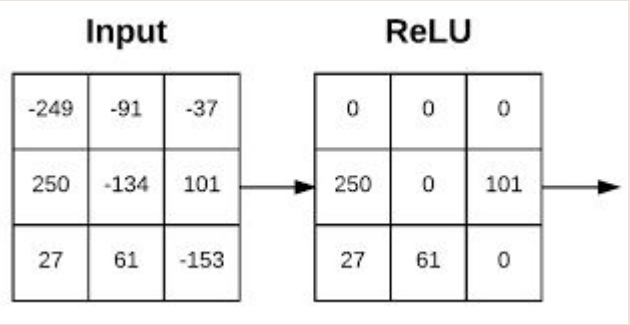
- First layer of cnn
- Stores the pixelated values of image into an array
- Used for extracting the features of the image and reduces its dimensionality



2. ReLu (Activation function)
"Converts negative values into zero"

- Relu is a half rectifier
  f(y) = 0 when y < 0
  f(x) = y when y > = 0
- Range of ReLu : [0 to infinity]

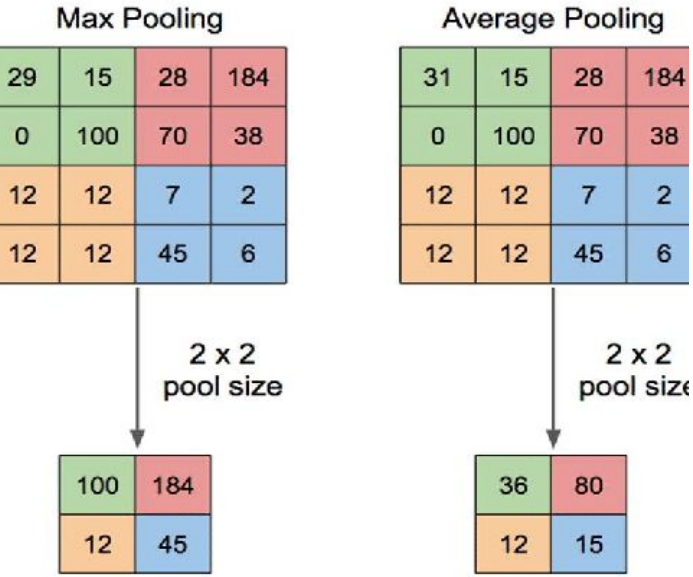




3.)POOLING LAYER

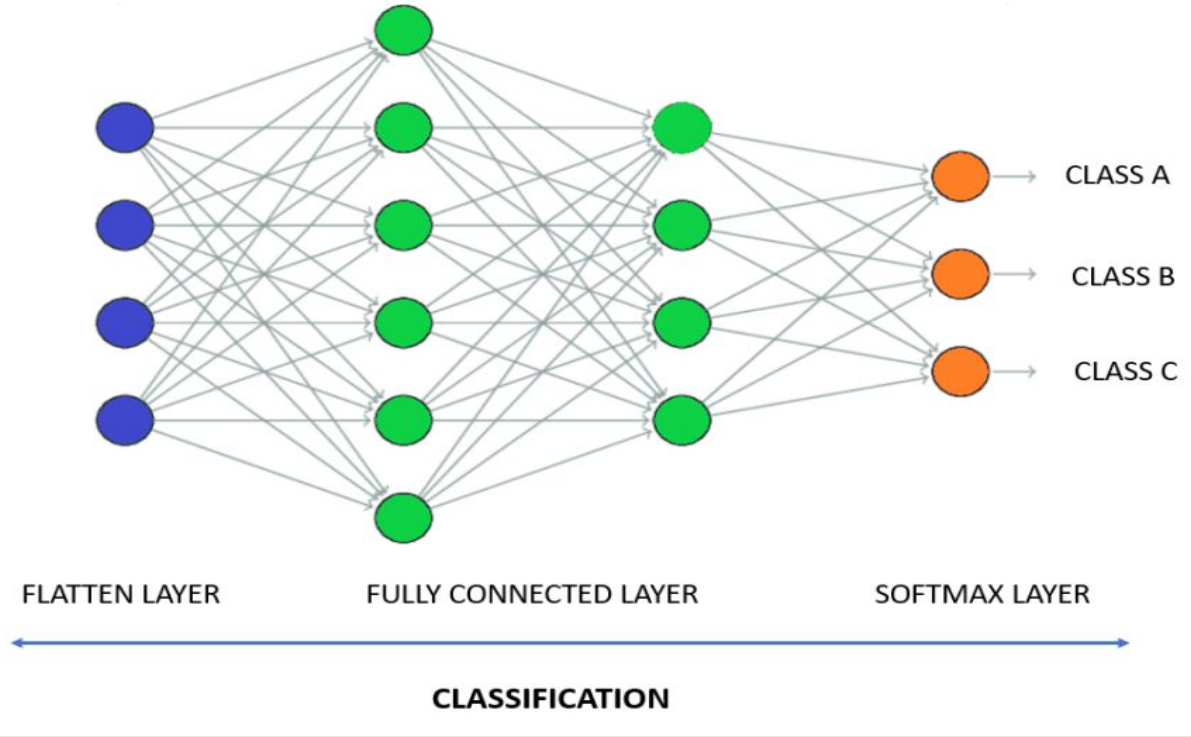
“reduces the spatial size and the numbers of parameters”

- o Used to reduce dimensionality
- o Helps to control overfitting
- o Filters of size 2x2 are commonly used in it



4.)FULLY CONNECTED LAYER

“Combines all the features together to create a final model



LIBRARIES IMPLEMENTATION

NUMPY **NumPy** is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several other developers. NumPy is open-source software and has many contributors.

PANDAS Pandas is a software library written for the python programming language for data manipulation and analysis. In particular, it offers data structure and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license.

MATPLOTLIB Matplotlib is a plotting library for the python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into application using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK. There is also a procedural “pylab” interface based on a state machine (like OpenGL), designed to closely resemble that of MATLAB, though its use is discouraged. SciPy makes use of Matplotlib

KERAS keras is an open-source software library that provides a Python interface for artificial intelligence neural neural networks. Keras acts as an interface for the Tensorflow library.

OPENCV (Open source computer vision library) is a library of programming functions mainly aimed at real-time computer vision. The library is a cross-platform and free for use under the open-source Apache 2 License. Starting with 2011, OpenCV features GPU acceleration for real-time operations.

OS The OS module in Python is a part of the standard library of the programming language. When imported, it lets the user interact with the native OS Python is currently running on. In simple terms, it provides an easy way for the user to interact with several os functions that come in handy in day to day programming.

WARNINGS Warnings are provided to warn the developer of situations that aren't necessarily exceptions. Usually, a warning occurs when there is some obsolete of certain programming elements, such as keyword, function or class, etc. A warning in a program is distinct from an error. Python program terminates immediately if an error occurs. Conversely, a warning is not critical. It shows some message, but the program runs. The warn() function defined in the 'warning' module is used to show warning messages.

INTRODUCTION TO TENSORFLOW

TensorFlow is a free and open source software library for machine learning and artificial intelligence. It can be used across a range of tasks
but has a particular focus on training and inference of deep neural networks.

TensorFlow

Losses

To train and assess models, TensorFlow provides a set of loss functions (also known as cost functions). Some popular examples include mean squared error (MSE) and binary cross entropy (BCE). These loss functions compute the “error” or “difference” between a model’s output and the expected output (more broadly, the difference between two tensors). For different datasets and models, different losses are used to prioritize certain aspects of performance.

Metrics

In order to assess the performance of machine learning models, TensorFlow gives API access to commonly used metrics. Examples include various accuracy metrics (binary, categorical, sparse categorical) along with other metrics such as Precision, Recall, and Intersection-over-Union (IoU).

TF.nn

TensorFlow.nn is a module for executing primitive neural network operations on models. Some of these operations include variations of convolutions (1/2/3D, Atrous, depthwise), activation functions (Softmax, RELU, GELU, Sigmoid, etc.) and their variations, and other Tensor operations (max-pooling, bias-add, etc.).

Optimizers

TensorFlow offers a set of optimizers for training neural networks, including ADAM, ADAGRAD, and Stochastic Gradient Descent (SGD). When training a model, different optimizers offer different modes of parameter tuning, often affecting a model’s convergence and performance.

IMPORTED LIBRARIES FROM TENSORFLOW

Categorical cross entropy: Categorical cross entropy is a loss function that is used in multi-class classification tasks. These are tasks where an example can only belong to one out of many possible categories, and the model must decide which one. Formally, it is designed to quantify the difference between two probability distributions.
The categorical crossentropy is well suited to classification tasks, since one example can be considered to belong to a specific category with probability 1, and to other categories with probability 0.

Example: The MNIST number recognition tutorial, where you have images of the digits 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9.
The model uses the categorical cross entropy to learn to give a high probability to the correct digit and a low probability to the other digits.

Activation functions

Softmax is the only activation function recommended to use with the categorical crossentropy loss function. the output of the model only needs to be positive so that the logarithm of every output value exists. However, the main appeal of this loss function is for comparing two probability distributions. The softmax activation rescales the model output so that it has the right properties.

Metrics

A metric is a function that is used to judge the performance of your model.Metric functions are similar to loss functions, except that the results from evaluating a metric are not used when training the model. Note that one may use any loss function as a metric.

Image data generator
The image augmentation technique is a great way to expand the size of your dataset. one can come up with new transformed images from your original dataset. But many people use the conservative way of augmenting the images i.e. augmenting images and storing them in a numpy array or in a folder.

Mobilenet
MobileNet is a type of convolutional neural network designed for mobile and embedded vision applications. They are based on a streamlined architecture that uses depth wise separable convolutions to build lightweight deep neural networks that can have low latency for mobile and embedded devices.

Early stopping
In machine learning, early stopping is a form of regularization used to avoid overfitting when training a learner with an iterative method, such as gradient descent. Such methods update the learner so as to make it better fit the training data with each iteration.

modelcheckpoint
ModelCheckpoint callback is used in conjunction with training using model.fit() to save a model or weights (in a checkpoint file) at some interval, so the model or weights can be loaded later to continue the training from the state saved.

A few options this callback provides include:

- Whether to only keep the model that has achieved the "best performance" so far, or whether to save the model at the end of every epoch regardless of performance.
- Definition of 'best'; which quantity to monitor and whether it should be maximized or minimized.
- The frequency it should save at. Currently, the callback supports saving at the end of every epoch, or after a fixed number of training batches.
- Whether only weights are saved, or the whole model is saved.

Data set source :
from dropbox
link:https://www.dropbox.com/s/nilt43hyl1dx82k/dataset.zip?dl=0

Building of the model

So we were working with pre trained model which was MobileNet
With standard dimensions values 224, 224, 3 and false include_top_layer, weights were given
And we used functional API imported model from keras.models

Again , we included 7 flatten and dense layer with activation Function of value softmax

Then we used optimizer “adam” for neural network training and for loss it was categorical_crossentropy since we used softmax as activation function

So we used image data generator for train and to generate image from our datasets

Using Image data generator we set a zoom range otherwise model to be able to read our data accurately

Then we imported our dataset with target size of 224, 224 and batch size of 32 (it will rerun our whole dataset 32 times for training)

Since we included only 7 layers so our whole data gets divided into 7 categories with 7 emotions (happy , fear, sad, surprise, etc.)

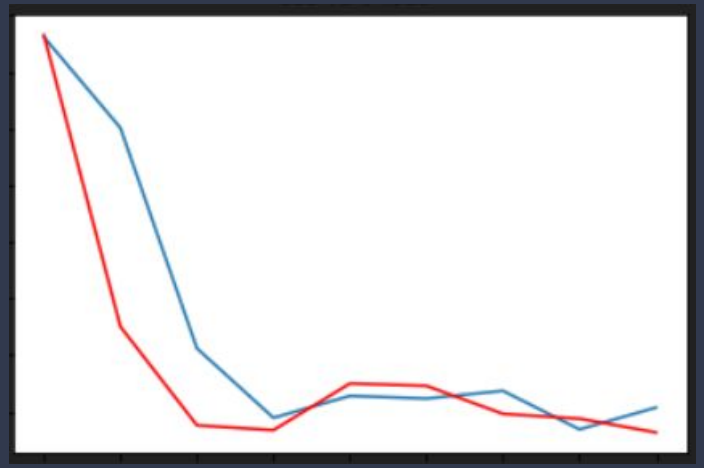
For validation and testing again values were same as training.

Then we generated our model with name best_model.h5

To check the accuracy and correct emotion classification
We used jpg picture from google and the result was almost accurate

Loss vs Validation loss

So validation loss after 9th epochs was 5.26 and loss was 6.14.
No improvement from previous val_loss 0.43
After plotting the graph difference between both were less



Accuracy vs validation accuracy

Accuracy after 9th epochs was 0.4125 and validation accuracy was 0.43
No improvement from previous val_accuracy 0.43
After plotting the graph both lines were approximately similar. Hence, model was stable

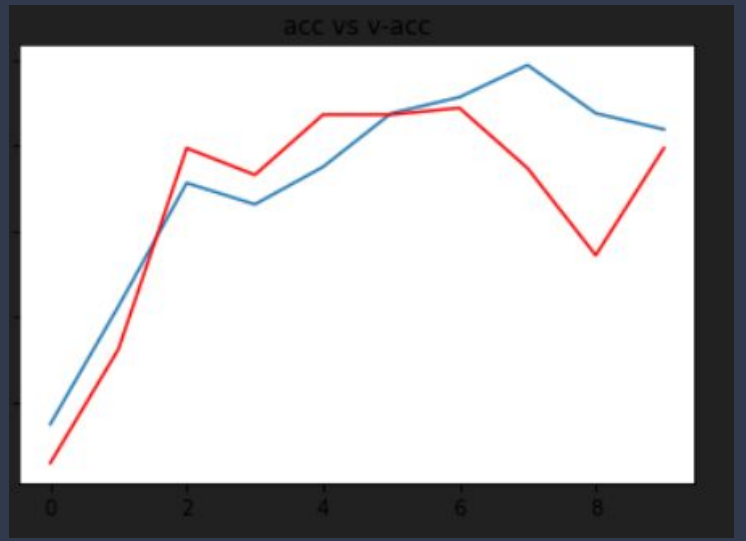


IMAGE PROCESSING

Image processing is a method to perform some important operations on an image. In order to get an enhanced quality image or to extract the most useful important from that

- **It is a one type of signal processing**
- **In this processing input is an image and output may be image or characteristics/features associated with that image**

How image input works in computer?

Converts the image into an array of
Pixel values where the dimension of array
Depends on the resolution of the image

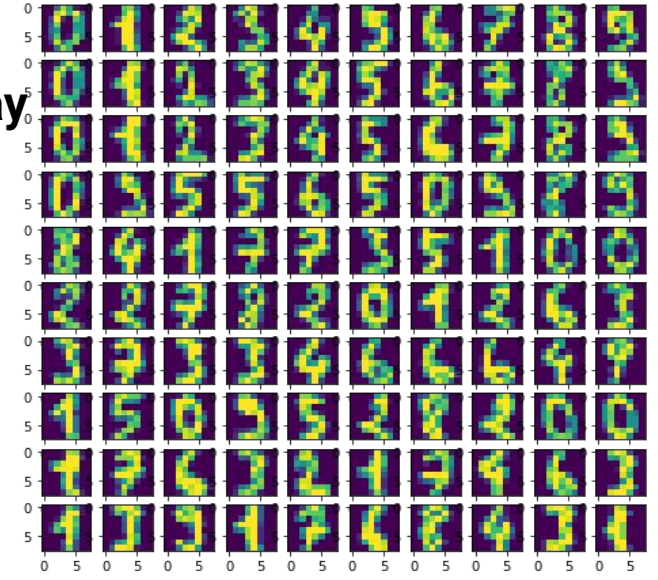


Image processing

Real and Visualize IMages using opencv

Image filters

Video processing using opencv

Face detection and recognition

IMAGE FILTERING

Image filtering is the process of modifying an image by changing its shades or colour of the pixel. It is also used to increase brightness and contrast.

LAYER PADDING

Padding is a term relevant to convolutional neural networks as it refers to the amount of pixels added to an images when it is being processed by the kernel or filter

IMAGE THRESHOLD

The threshold is converted everything to white or Black, based on the threshold value.

CONNECTED COMPONENTS

Connected components labelling scans an image and groups its pixels into components based on pixel connectivity, i.e, all pixels In a connected component share similar pixel intensity values and are in some way connected with each other.

Face recognition using opencv

The face recognition is a technique to identify or verify the face from the digital images or video frame. A human can quickly identify the faces without much effort. It is an effortless task for us, but it is a difficult task for a computer. There are various complexities, such as low resolution, occlusion, illumination variations, etc. These factors highly affect the accuracy of the computer to recognize the face more effectively. First, it is necessary to understand the difference between face detection and face recognition.

Face Detection: The face detection is generally considered as finding the faces (location and size) in an image and probably extract them to be used by the face detection algorithm.

Face Recognition: The face recognition algorithm is used in finding features that are uniquely described in the image. The facial image is already extracted, cropped, resized, and usually converted in the grayscale.

Introduction to opencv

Computer Vision

Computer vision is a process by which we can understand the images and videos how they are stored and how we can manipulate and retrieve data from them. Computer Vision is the base or mostly used for Artificial Intelligence. Computer-Vision is playing a major role in self-driving cars, robotics as well as in photo correction apps.

INTRODUCTION TO OPENCV

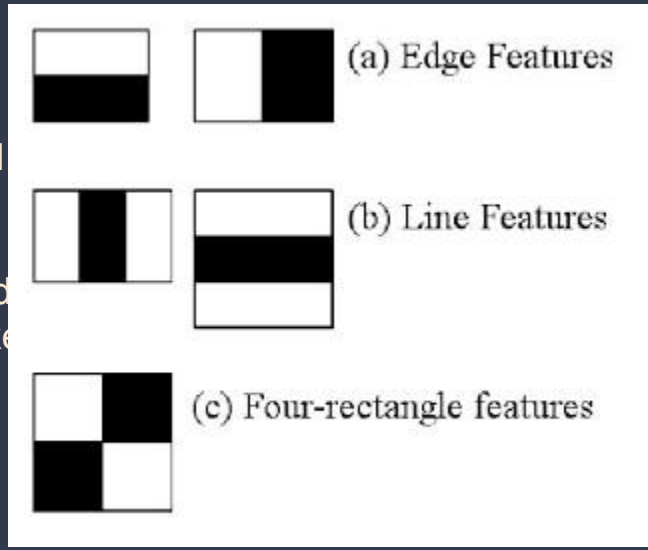
```
import numpy as np
import cv2

cap = cv2.VideoCapture(0) # it will capture video from webcam

while True:
    ret, frame = cap.read()

    cv2.imshow('frame', frame) # title of the frame will be frame

    if cv2.waitKey(1) == ord('q'): # key q help to close the frame
        Break                      # and escape the window

cap.release()
cv2.destroyAllWindows()   # destroys other window
```

OUTPUT: Webcam turned on capturing the capturing the video

Haarcascade_frontalface_default.xml

Create the Matrix with a Pixel value range from 0 to 255 (0 White and 255 Black)Pixel is the smallest element of an image. Each pixel corresponds to anyone value. The value of the pixel is between 0 and 255. The value of a pixel at any point corresponds to the intensity of the light photons striking at that point.

Object Detection using Haar feature-based cascade classifiers is an effective method proposed by Paul Viola and Michael Jones in the 2001 paper, "Rapid Object Detection using a Boosted Cascade of Simple Features". It is a machine learning based approach in which a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images.
Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it. For this, Haar features are used. They are just like our convolutional kernel. Each feature is a single value obtained by subtracting the sum of pixels under the white rectangle from the sum of pixels under the black rectangle.



Emotion detection of Real time video

Detecting the real-time emotion of the person with a camera input is one of the advanced features in the machine learning process. The detection of emotion of a person using a camera is useful for various research and analytics purposes. The detection of emotion is made by using the machine learning concept.

So we again implemented our best_model.h5 which

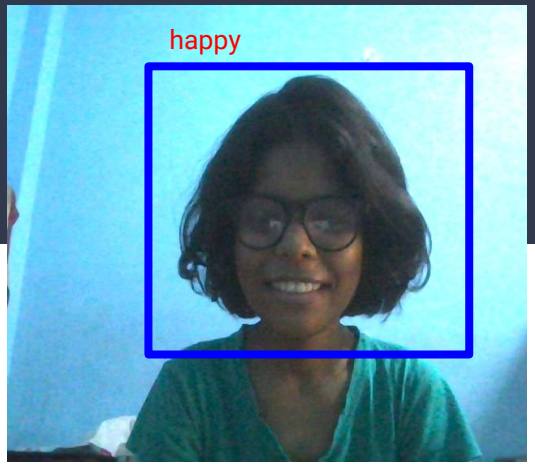
Was Trained earlier into real time emotion detection using haarcascade_frontalface algorithm

And the accuracy wasn't great .

Our model was able to classify some emotions such as happy, fear, neutral but not all the 7 emotions

So we weren't able to capture all emotion as it was rarely capturing some specific emotion . so we put some dummy examples to show how it captured and it was quite similar.

Results



Review