FACE EMOTION RECOGNITION USING CNN

Team Members:	Angana Banerjee (521) Shreya Banerjee (520) Pratyusha Mitra (554)
Supervisor:	Professor Shalabh Agarwal
Department:	M.Sc. Computer Science (MCMS)
College:	St. Xavier's College(Autonomous)**,** Kolkata

SYNOPSIS

Objective/Aim: -

The main aim of our project is to detect the frontal face and its underlying emotions it contains using HAAR CASCADES classifier and CNN. The first step is to detect the frontal face from the real time video and crop the region of interest and then detection of emotion pattern is done and then there is a need to discard the non-face images as early as possible to have better accuracy.

Technical details: -

The project design can be divided into three sections: -

1) CNN Model
2) Database
3) User Interface

The challenging aspect our project lies on the fact--understanding of emotions. For this reason, we are using a deep learning model--CNN for recognising emotions in an accurate way. CNN model performs in such a way that it can helps in the emulation of visual cortex system in human. CNN contains individual neurons which are connected in such a way that it learns from restricted regions in the image which in turn overlap each other to create connectivity in the entire image. CNN consists of input, hidden and output layers and hidden layer again is comprised of convolutional layers, max pooling layers and fully connected layers so presence of all these makes CNN exceptionally good in extraction of features from an image and learning from those features on its own. The CNN, like any other deep learning model, consists of 2 stages, Forward Propagation and Backward Propagation.

1. *Forward Propagation*: the model inputs image, extract features, processes an output.
2. *Backward Propagation*: the loss is computed and fed back into the network to correct parameters in the network.

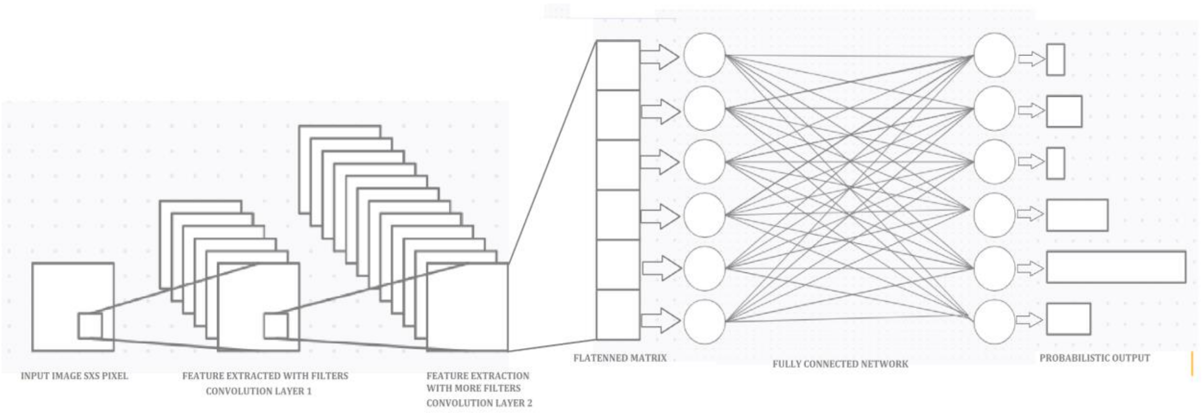
INPUT IMAGE SXS PIXEL
FEATURE EXTRACTED WITH FILTERS CONVOLUTION LAYER 1
FEATURE EXTRACTION WITH MORE FILTERS CONVOLUTION LAYER 2
FLATENNED MATRIX
FULLY CONNECTED NETWORK
PROBABILISTIC OUTPUT

Fig. 1 -Typical CNN architecture

Materials and requirements: -

HARDWARE USED: -

- ✓ **RAM** — 8GB DDR4
- ✓ **Processor** — i5 or AMD RAZON 5 / INTEL(R)-CORE(TM) i5-8265U CPU @1.60GHz 1.80GHz
- ✓ ***Graphics card*** ---- INTEL UHD GRAPHICS 620

SOFTWARE USED: -

- ✓ **IDE** — Jupyter Notebook/Anaconda
- ✓ **Libraries used** — Numpy, Seaborn, Keras, Matplotlib, Scikit-learn, OpenCV
- ✓ **Dataset** — Facial Expression Images Dataset
- ✓ **Programming language** — Python

Database Used: -

For our project, we are using a Facial Expression Images Dataset containing 35887 images and 7 emotions. The dataset is split into training and validation in the ratio 4:1 (approximately).

The following table shows the number of images for each emotion in the training and testing dataset: -

S. No	Emotions	Training	Validation	Total
1	Happy	7164	1825	8989
2	Neutral	4982	1216	6198
3	Fear	4103	1018	5121
4	Surprise	3205	797	4002
5	Angry	3993	960	4953
6	Disgust	436	111	547
7	Sad	4938	1139	6077

Model Specification: -

The architecture of our model can be visualised as given below: -

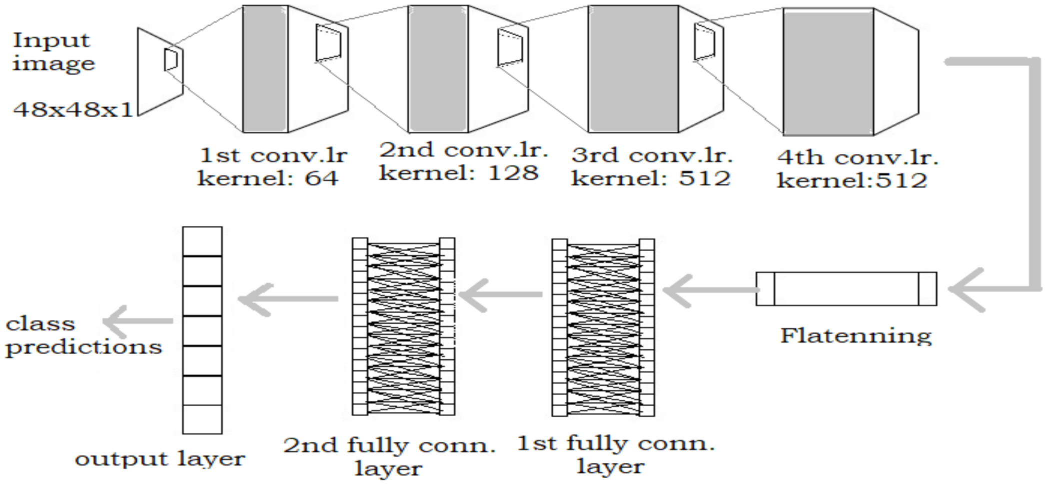

Fig. 2 – CNN Architecture of our Project

Our model consists of 4 convolution layers and 2 fully connected layers.

- The first convolution layer or the input layer is supposed to take an input of dimension 48×48×1. This layer is supposed to have the following attributes.
    1) No of kernels - 64

2) Filter size - 3×3
3) Pooling (Max pooling) - 2×2
4) Activation function used - ReLU
5) Drop out - 0.25

- In the second convolution layer the input size is kept the same as the output of the first convolution layer. The second convolution layer would have the following attributes.
    1) No of kernels - 128
    2) Filter size - 5×5
    3) Pooling (Max pooling) - 2×2
    4) Activation function used - ReLU
    5) Drop out - 0.25
- In the third convolution layer the input size is kept the same as the output of the first convolution layer. The second convolution layer would have the following attributes.
    6) No of kernels - 512
    7) Filter size - 3×3
    8) Pooling (Max pooling) - 2×2
    9) Activation function used - ReLU
    10) Drop out - 0.25
- In the fourth convolution layer the input size is kept the same as the output of the first convolution layer. The second convolution layer would have the following attributes.
    11) No of kernels - 512
    12) Filter size - 3×3
    13) Pooling (Max pooling) - 2×2
    14) Activation function used - ReLU
    15) Drop out - 0.25
- We introduce a flatten layer which essentially flattens the output of the fourth convolution layer (3d matrix) into a 1-D matrix.
- Next, the flattened matrix is fed into the fully connected layers The first fully connected layer consists of the following attributes.
    1) No. of kernels - 256
    2) Activation function - ReLU
    3) Drop out - 0.25
- The second fully connected layer consists of the following attributes.
    4) No. of kernels - 512
    5) Activation function - ReLU
    6) Drop out - 0.25
- Finally, a dense layer is added as the output layer containing the following attributes.
- Activation function - Softmax
- No of classes - 7 (No of emotions recognized in phase 1)
- Additionally, the model is optimized with Adam having the initial LR (Learning rate) as 0.0001

- Finally, the model is compiled with loss function categorical cross entropy, penalizing itself on metrics of accuracy.

Depending upon the size of data, the number of epochs can be set to 50. After training the model with the training dataset, to files would be generated as follows:

1. Model_weights.h5
2. Model as model. Json

These files generated are used in the next phase that is on the implementation section.

User Interface: -

In brief the steps are:

- ✓ Capture of image frame from real time video with the help of USB camera
- ✓ Pre-processing of the images received
- ✓ The CNN model will extract all the important features from image frame
- ✓ Use of HAAR-CASCADE to detect the person’s face(frontal) in particular
- ✓ Usage of the trained deep learning model to detect the underlying emotions of the face that is captured like happy, sad, neutral, fear, disgust, angry or surprised
- ✓ Display of those emotions detected from the real time video is then being seen by the user

Design flow: -

The flow of the process is shown with the help of the flowchart in the next page (6): -

Now this can be a complex task due to the following reasons:

1. This detects emotions on the basis of face not on the contexts.
2. This detects on 7 emotions as of now.
3. Only the frontal face is taken into consideration.
4. Emotions are highly subjective in nature and its interpretation also varies from person to person and it’s hard to define its notion sometimes.

Fig. 3 -The flow of process.

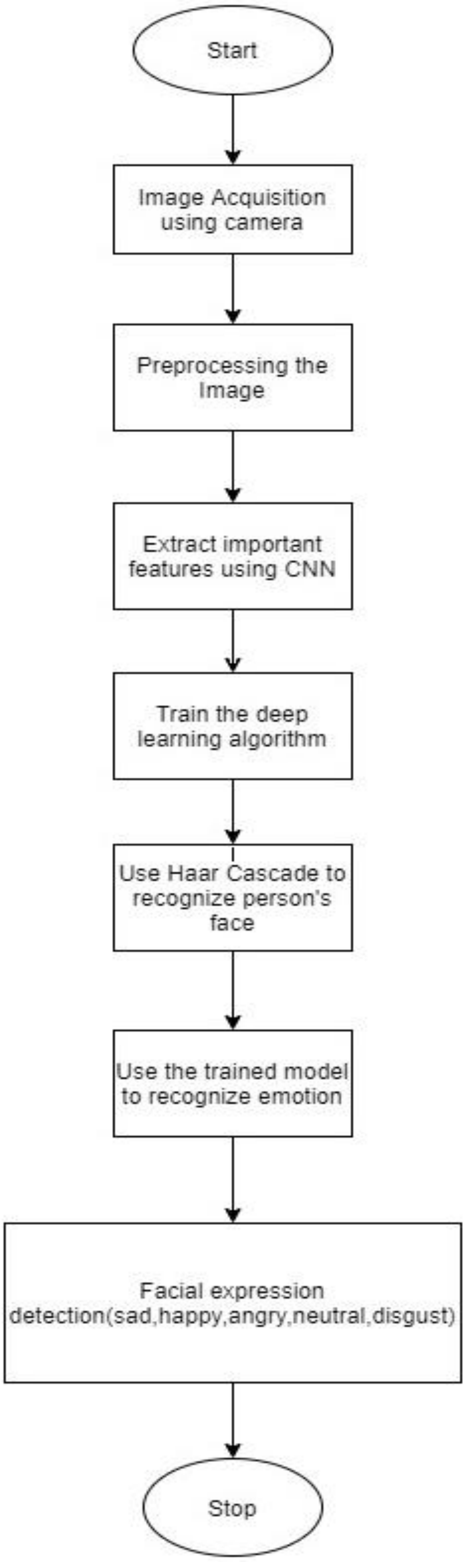
Start
Image Acquisition using camera
Preprocessing the Image
Extract important features using CNN
Train the deep learning algorithm
Use Haar Cascade to recognize person's face
Use the trained model to recognize emotion
Facial expression detection(sad,happy,angry,neutral,disgust)
Stop

Fig. 3 -The flow of process.