**Optimal Facial Feature Based Emotional Recognition Using Deep Learning Algorithm**

**Introduction:**

Emotional recognition is one of the most crucial and challenging techniques nowadays. Emotion recognition is used in a wide range of applications, like helping to evaluate blood pressure, stress levels, etc. Our key objective here is to create a AI model that can detect the following emotions like happy, sad, surprise, neutral, fear, horror, cry from the image or the video input.

**Model Presented in the research paper:**

FER 2013 dataset which is present in the Kaggle community website is taken for this research. The dataset contains the various expressions of the facial image. This input goes through the preprocessing technique. The two primary purposes of the preprocessing technique are to 1) add a filter to remove noise and 2) convert an RGB image to a greyscale image. Gaussian filter is used to reduce the noise in the input image and smoothen the edges. The next stage after the preprocessing is the Feature extraction stage. Feature extraction is one of the most crucial techniques in image processing. The canny filter is mainly used for edge detection in the image. The first step for the canny filter is to detect the noise with the Gaussian filter. The gradient magnitude helps smoothen the image after the smoothening threshold process is involved to determine the potential edges. Once the edges are detected the image is converted to binary image by using the valuable threshold values. This feature extracted image is given as input through the convolutional neural network using the deep learning method. The model consists of two convolution layers with RELU activation followed by a Max pooling layers. Finally, two fully connected layers are used to get the prediction label.

**Fit falls in the Existing Model:**

Since the number of trainable parameters are more the training usually takes more time.

**Proposed Work:**

In our proposed work the normal convolutions present in the CNN model is replaced with depth wise convolution followed by point wise convolutions. This is done in order to reduce the number of trainable parameters. Also, for faster convergence we have used gradient clipping and learning rate decay. Also, in order to make our model to better understand the problem additional convolution layers are added.

**Conclusion:**

**As a result of the above changes the training time of the model is reduced and also more accuracy on validation dataset is achieved.**