**Face Emotion Recognition**

Jai Harish,

Ashik Kumar,

Pranil Thorat,

Bhaskar subanji,

Saransh Srivastava

Data science trainee,

Alma Better, Bangalore

**Abstract:**

* Recognition of facial expressions (FER) our objectives are twofold: we want to make a difference in the world, but we also want to make a difference not just in enhancing precision, but also applying our findings to the real world. In our study, we demonstrate state-of-the-art nearly 72 percent accuracy on the test using a variety of current research strategies. The validation set for FER2013 outperformed all previous publications. We also show off a web app that runs our FER. In real time, models are created on the device.

**1. Problem Statement**

* The aim of the project is to create a Facial Emotion Recognition System (FERS) that can detect students' emotional states in e-learning systems that use video conferencing.
* This technology instantly conveys the emotional states of the students to the educator in order to create a more engaged educational environment.
* Our results supported those of other studies that have shown that in e-learning systems, it is possible to observe the motivation level of both the individual and the virtual classroom.

**2. Introduction**

* The process of recognizing human emotions on the face is known as face emotion recognition. People's ability to recognize other people's emotions obviously varies. The use of technology to assist individuals in detecting emotions is a relatively new study field. In general, technology works best when various modalities are used in conjunction. Automating the recognition of facial expressions from video, spoken expressions from audio, written expressions from text, and physiology as measured by wearables has received the most attention to date.
* Nonverbal communication includes facial expressions. Several studies have been conducted to classify these facial expressions. The universal facial expressions of seven emotions, including neutral happiness, sadness, anger, disgust, fear, and surprise, have substantial evidence. As a result, detecting these emotions on the face is critical as it has a wide range of applications in the fields of computer vision and artificial intelligence. These domains are investigating facial expressions in order to automatically detect human attitudes.

**3. Steps involved:**

The following steps are involved in the project

1. **Data Gathering**:

We have downloaded the data from Kaggle (link -<https://www.kaggle.com/deadskull7/fer2013>). Dataset is in .csv format where we have emotions, pixels and usage as the columns. So, we loaded the dataset and performed data preprocessing on it.

1. **Data Augmentation:**

Image data augmentation is a technique that can be used to artificially expand the size of a training dataset by creating modified versions of images in the dataset.

Training deep learning neural network models on more data can result in more skilful models, and the augmentation techniques can create variations of the images that can improve the ability of the fit models to generalize what they have learned to new images.

The Keras deep learning neural network library provides the capability to fit models using image data augmentation via the Image Data Generator class.

1. **Splitting the data:**

We have split the data into training and testing with a test size of 20%

1. **Model Training:**

Mobilenet, Dexpression, CNN, Densenet, and ResNet were all used. After testing each model, it was discovered that ResNet delivered the best results of all the models used.

1. **Model Performance**

To evaluate how our model performed we have used classification report and confusion matrix

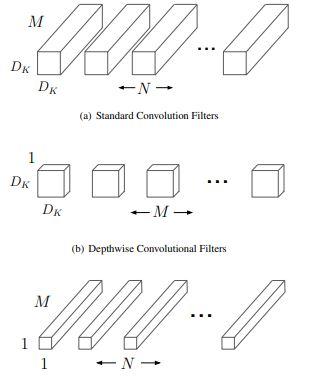
1. **Real time video detection using webcam:**

In order to test our model on a real time video that can be done using OpenCV library. When writing the code for detecting the face we have also used our finalized model to detect face emotions.

**4. Algorithm:**

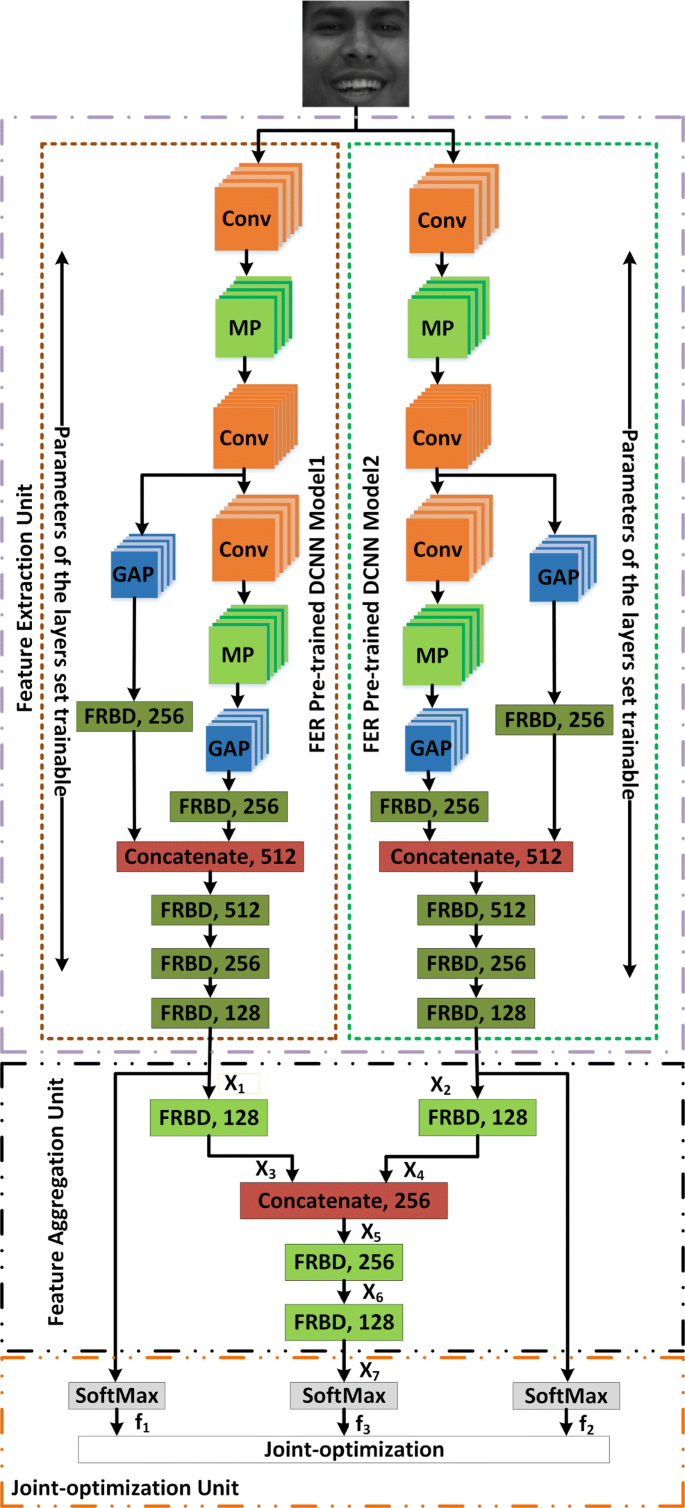
1. **Mobilenet:**

MobileNet is an efficient and portable CNN architecture that is used in real-world applications. MobileNets primarily use depth-separable convolutions in place of the standard convolutions used in earlier architectures to build lighter models. MobileNets introduces two new global hyperparameters (width multiplier and resolution multiplier) that enable model developers to trade off latency or accuracy for speed and low size based on their needs.



**2. Dexpression:**

The suggested architecture outperforms the current state of the art utilizing CNNs by 99.6 percent for CKP and 98.63 percent for MMI. Face recognition software has a wide range of applications, including human-computer interface and safety systems. This is because nonverbal cues are vital types of communication that play an important part in interpersonal interactions. The usefulness and dependability of the suggested work for real-world applications is supported by the performance of the proposed architecture.



**3. CNN:**

Basic CNN architecture details:

• **Input layer** - The input layer in CNN should contain image data.

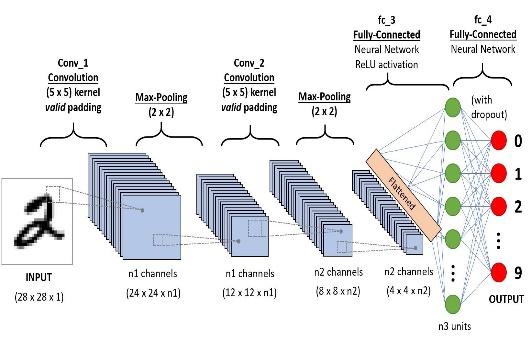
• **Convo layer** - The convo layer is sometimes called the feature extractor layer because features of the image are get extracted within this layer

• **Pooling layer** - Pooling is used to reduce the dimensionality of each feature while retaining the most important information. It is used between two convolution layers.

• **Fully CL** - Fully connected layer involves weights, biases, and neurons. It connects neurons in one layer to neurons in another layer. It is used to classify images between different categories by training and placed before the output layer

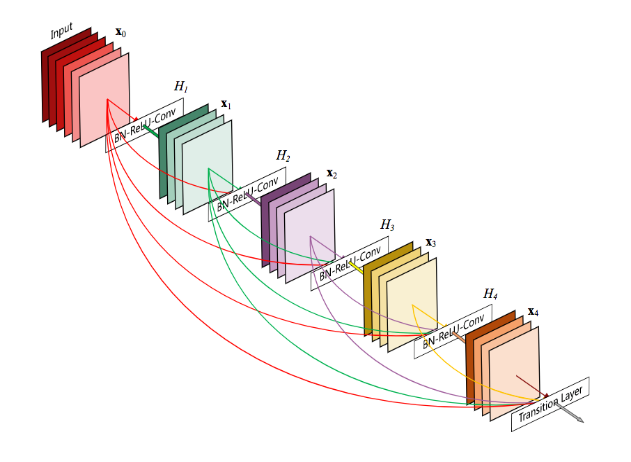
• **Output Layer** - The output layer contains the label which is in the form of a one-hot encoded.

A **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.



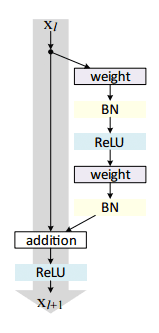
**5. Densenet**

DenseNet was developed specifically to improve the declining accuracy caused by the vanishing gradient in high-level neural networks. In simpler terms, due to the longer path between the input layer and the output layer, the information vanishes before reaching its destination.



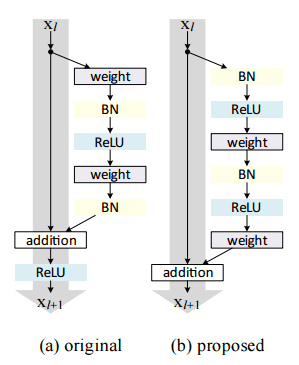
**5. ResNet:**

The term micro-architecture refers to the set of “building blocks” used to construct the network. A collection of micro-architecture building blocks (along with your standard CONV, POOL, etc. layers) leads to the macro-architecture (i.e., the end network itself). First introduced by He et al. in their 2015 paper, the ResNet architecture has become a seminal work, demonstrating that extremely deep networks can be trained using standard SGD (and a reasonable initialization function) through the use of residual modules:



Further accuracy can be obtained by updating the residual module to use *identity mappings*, as demonstrated in their 2016 follow-up publication,

[**Identity Mappings in Deep Residual Networks**](https://arxiv.org/abs/1603.05027):



* That said, keep in mind that the ResNet50 (as in 50 weight layers) implementation in the Keras core is based on the former 2015 paper.
* Even though ResNet is much deeper than VGG16 and VGG19, the model size is actually substantially smaller due to the usage of global average pooling rather than fully-connected layers — this reduces the model size down to 102MB for ResNet50.

**5. Model performance:**

##### **i) Confusion Matrix-**

The confusion matrix is a table that summarizes how successful the classification model is at predicting examples belonging to various classes. One axis of the confusion matrix is the label that the model predicted, and the other axis is the actual label.

##### **Precision, Recall, F1 score and Support-**

* Precision is the ratio of correct positive predictions to the overall number of positive predictions: **TP/TP+FP**
* Recall is the ratio of correct positive predictions to the overall number of positive examples in the set: **TP/FN+TP**
* It is also called the F Score or the F Measure. Put another way, the F1 score conveys the balance between the precision and the recall The F1 Score is the **2\*((precision\*recall)/(precision+recall))**
* Support is **the number of actual occurrences of the class in the specified dataset**. Imbalanced support in the training data may indicate structural weaknesses in the reported scores of the classifier and could indicate the need for stratified sampling or rebalancing

**ii) Accuracy and loss curve-**

* Accuracy is **a method for measuring a classification model's performance**. It is typically expressed as a percentage. ... Accuracy is often graphed and monitored during the training phase though the value is often associated with the overall or final model accuracy. Accuracy is easier to interpret than loss.
* Loss value implies how poorly or well a model behaves after each iteration of optimization. An accuracy metric is used to measure the algorithm's performance in an interpretable way. It is **the measure of how accurate your model's prediction is compared to the true data.**

**6. Model Deployment-**

**Creating Web App Using Streamlit-**

* Streamlit is a Python framework for developing machine learning and data science web apps that is open-source. Using Streamlit, we can quickly create web apps and deploy them. You can use Streamlit to make an app the same way you'd make a Python programme. It's possible with Streamlit. Working on the interactive loop of coding and viewing results is a pleasure. In the web application.

**Deployment in cloud platform-**

* AWS (Amazon Web Services) is a comprehensive, evolving cloud computing platform provided by Amazon that includes a mixture of infrastructure as a service (IaaS), platform as a service (PaaS), and packaged software as a service (SaaS) offerings.

**Deployment Link for AWS-**

<http://15.206.194.193:8501/>

1. **Conclusion:**

* All the models of Mobilenet, Dexpression, CNN, Densenet, and ResNet were evaluated.
* The ResNet model was chosen because it had the highest training accuracy of all the models, and its validation accuracy was nearly 72 percent, which is comparable to CNN models.
* Other models, such as Mobilenet, Dexpression, and Densenet, didn't give the appropriate result.
* As a result, we save this resnet model and use it to predict facial expressions.
* Using streamlit, a front-end model was successfully created and ran on a local webserver.
* The application is able to detect face location and predict the right expression while checking it on a local webcam.