**IMPLEMENTATION**

Implementation phase is the actual building of the system usually referred to as Coding phase. The system architecture and data flow from the previous phase are referred in this phase so as to be consistent with the design document.

**Tools and Technology Introduction**

The tools and technologies deployed in the project are:

**Python:** Python is a high-level programming language used for passing and editing the dataset. It is the fundamental language behind all the tools we use for deep learning. The python libraries such as FFmpeg and Imagemagick are used for frame generation. It also consists of many in-built libraries such as Theano, Keras that are favorable for deep learning, thereby making the life of programmer easier.

**Keras:** Keras is a python based deep learning library that allows easy and fast modeling with high modularity and minimal code, capable of running on top of either TensorFlow or Theano. It supports CNN, RNN and combination of both networks and can run efficiently on CPU as well as GPU. The API’s provided by keras are very simple and user friendly, where one can define powerful deep learning models in a very efficient manner.

**Caffe:** Caffe is a deep learning framework similar to PyLearn2, wherein the basic code is written in very highly Optimized C++ language by considering the time and space complexity. Caffe is the best choice for beginners to implement the standard algorithm. It is difficult to customize the existing model as it is very hard to reform the C++ code.

**Theano:** Theano is a deep learning numerical computation library based on python, which allows to process mathematical equations containing multi-dimensional arrays in a very effective manner. The syntax is similar to NumPy, and can be executed consistently on either CPU or GPU. It generates the C code dynamically to execute the code faster, where the developer has full control over the modeling of neural network.

**CuDNN:** The NVIDIA CUDA Deep Neural Network (cuDNN) is a GPU-accelerated library for deep neural networks, where the focus is mainly laid on designing and training neural network than on performance tuning. The upgraded version of cuDNN library is highly optimized and has been tested by NVIDIA for the implementation of CNNs. Caffe with cuDNN allows programmers to define models in plain text, thereby achieving the computational speedups.

**TensorFlow:** TensorFlow is a library for numerical computation using data flow graphs developed by Google, written in C++, wherein the nodes represent mathematical operations and the edges serve as the multi-dimensional data arrays (tensors). It has very flexible architecture that utilizes computation to one or more CPUs or GPUs across multiple devices with a single API. It can also work with any gradient-based machine learning algorithm. This is an alternative for the Theano library for the Keras to run.

**FFmpeg:** FFmpeg is an open source software libraries and programs to manage data related to multimedia, which has the capability to transcode, encode, decode, mux, demux, filter, stream etc.

**Bootstrap:** Bootstrap is an front-end library and a framework for HTML, JS and CSS in-order to develop a very user friendly, responsive and dynamic web-sites. It acts as an interface for the user, unlike the server-side code that resides on the back end (server).

**IPython:** An IPython notebook allows a user to write and execute Python code in web browser, making it easy to modify and execute in short snippets. IPython has an interactive python shell and architecture for parallel computing that multiple clients to connect to the web-based notebook.

**Socket programming:** Sockets acts as a gateway to communicate across two systems using TCP protocol. A socket is created at the client end which then connects to a server. The server creates a corresponding socket object when the connection is established, which is utilized for the communication using I/O streams.

**Common Gateway Interface (CGI):** To generate web pages dynamically, CGI acts as a standard way for web servers to interface with executable programs installed on server. CGI extends http(s) server software system by allowing the owner of the Web server to nominate a directory within the document collection as it contains executable scripts instead of pre-written pages; this is known as a CGI directory. Information can be sent via web browser to the script via the URL or an HTTP POST request.

**Overall Implementation View of the Project**

A video that is a feedback of a person responding to an advertisement or a movie, is captured using any of the video capturing tools such as web-cam and is uploaded from a web page to calculate the extent of expressiveness of the person for each second. The video that uploaded is chopped down into frames. Prior to this, a model is prepared by training the neural network with the available dataset using deep learning concepts such as CNN and LSTM. The frames are then given as input to these models that outputs a file comprising of the extent of expressiveness for each of the frame. The output file is then modified and integrated to obtain the expressiveness per second and stored in the local system as well as displayed on the web page and also line charts are plotted to make the output look better.

**Explanation of Algorithm and its Implementation**

The critical unit in the working of this project is the modules as specified in the design document, wherein the CNN and LSTM concepts of Deep Learning are made to the best use in the detection of extent of expressiveness of a person.

The brief algorithmic steps carried out for training purpose are illustrated below:

1. Loading of data (videos) into memory.
2. Pre-processing of data.
   1. The input videos are chopped down into frames, made possible by FFmpeg.
   2. These frames are adjusted to correct resolution (preferred to be 320x240).
   3. Transposing of frames or images if they are inverted.
   4. The frames are converted to grey scale image (necessary for LSTM model).
   5. Converting the frames to float by normalizing which is done by dividing by 255.
3. Splitting of data.
   1. A 70 - 30 split with 70% of the data used for training and the other 30% for cross validation.
4. Define and Configuration of Neural Networks.
   1. CNN
      1. 2 layers of network with the first layer consisting of 96 kernels, each of size 5x5x3 scans an image, generating array of data frames. These data frames are further sub-sampled using max pooling of size 2x2. Second layer consists of a 2D convnet of 32 kernels; each of size 3x3x3, which are further sub-sampled using max pooling of size 2x2 again.
      2. A Dropout of 25% is added to the output of above sub-sampling so as to avoid over-fitting. Dropouts help in regularizing the network and thereby increasing accuracy.
      3. The output after the dropouts are flattened and fed into a dense layer of *sigmoid* activation function.
      4. To determine the loss value, Mean Squared Error (MSE) is used with Adadelta as optimizer as it has adaptive learning rate.
   2. LSTM
      1. A LSTM is defined using keras library functions. The LSTM takes a gray scale image and scans it column by column. Since the image is rescaled to 80 x 60, the output of the LSTM model is a vector of size 80. In order to tackle the problem of vanishing gradients the hidden gate biases is fixed to a value of 1.
      2. A Dropout of 25% is added to the output as to avoid over-fitting.
      3. The output after the dropouts are fed into a *relu* activation function and MSE is used to determine the value loss along with Adadelta as optimizer.
5. With the help of Neural networks configured in step 4 and the 70% of the dataset, the neural networks are trained to evaluate the extent of expressiveness of a person.
6. The network is cross validated with the other 30% of the dataset and a model with weights is obtained as the output which is helpful in predicting the extent of expressiveness in other videos.

**Information about the Implementation of Modules**

The critical modules of the Implementation phase are

1. Training module

The Neural Network is configured and trained by a python script with the help of libraries such as Keras and Theano which are essential libraries for the usage of deep learning concepts. The Nvidia cuDnn is also used so as to process the data faster and get the results much quicker by the use of GPU rather than using CPU which takes more memory space as well as time to process. A model and weights are obtained as an output from the training module and are stored in a hdf5 file format. These files are used for the evaluation of expressiveness of other videos.

1. Web-based module

A web based GUI is provided for the user where he/she can upload a video. The expressiveness of that video is calculated based on the models and that which are obtained from the training module. A python script is called for the evaluation of extent of expressiveness when a video is uploaded which fragments the videos into frames and these frames are then given as input to the weights and models. The output is stored in a file as well as displayed on the web page.

1. Iot module

A raspberry Pi is used to trigger a web-cam to record a video and at the end of the recording, the video is transferred on to the system comprising of both the above mentioned modules with the help of socket programming. Here, the raspberry pi will act as a client and the system that receives the file will act as a server. All the calculations that mentioned above are done on the server side and the result is stored in a *.txt* file on the local system itself.

**Conclusion**

The overall system design as highlighted in design document is implemented with the help of Deep Learning concepts. A raspberry pi board is used to record a video through a web-cam and a web page is designed to upload the videos. The uploaded videos are graded on the extent of expressiveness of the subject in the video.