Psychometric Analysis Tool

Submitted by:
Apurva Bhargava (BE/25022/15)
Pallavi Jain (BE/25001/15)

Under mentorship of:
Dr. Shripal Vijayvargiya
Associate Professor, BIT Jaipur

Introduction

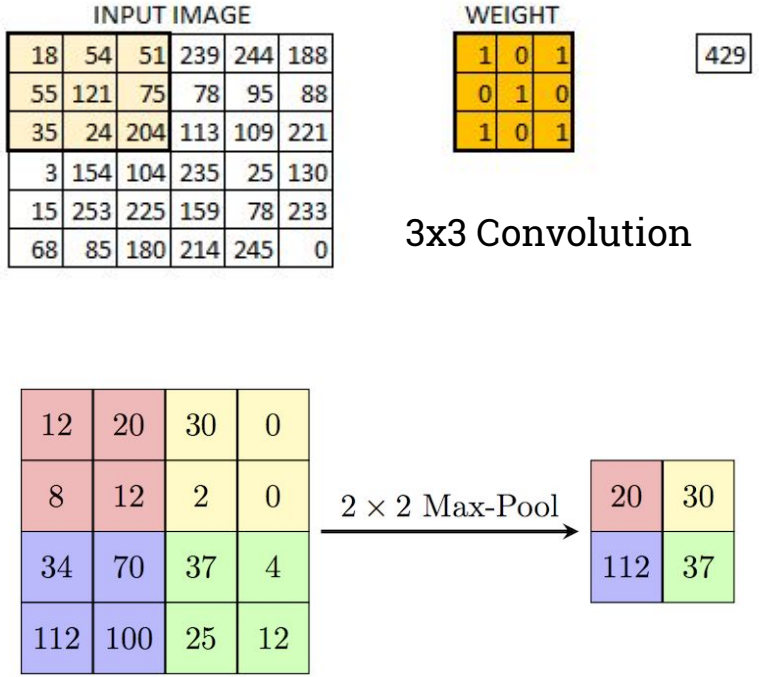
Psychometric Analysis Tool is a tool for interviewers and/or psychologists that powers judgement by determining the interviewee's psychological characteristics based on the detection and analysis of factors such as facial expressions, speech emotions, polarity of written opinion and question answering through chatbot interview.

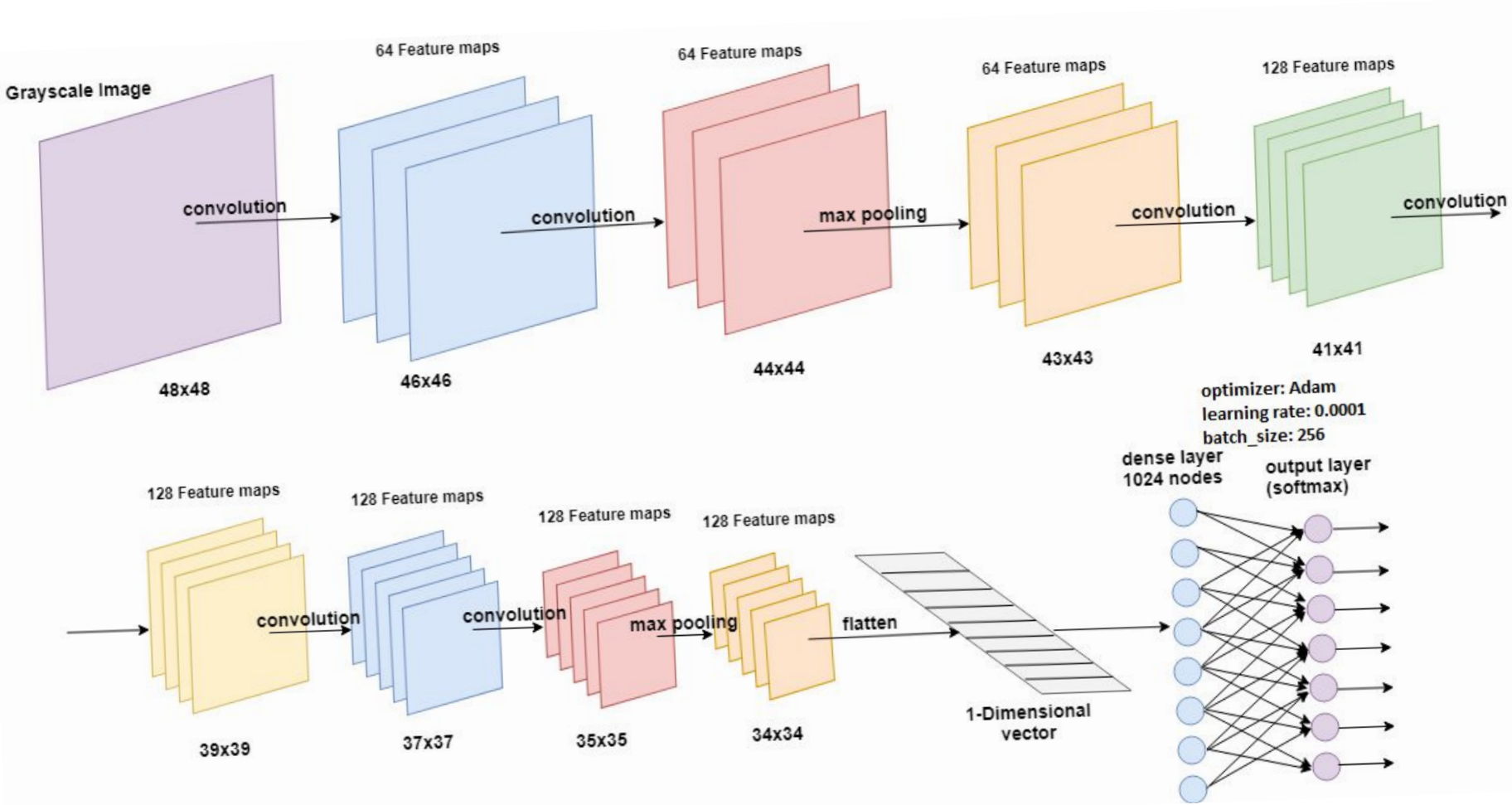
Objectives

- Facial expression recognition using images from video feed/ webcam
- Speech emotion recognition using audio feed
- Confidence and certainty scores of speech converted to text
- Sentiment analysis of image-based description
- Adaptive interviewing chatbot
- Browser User Interface, Frontend and Backend

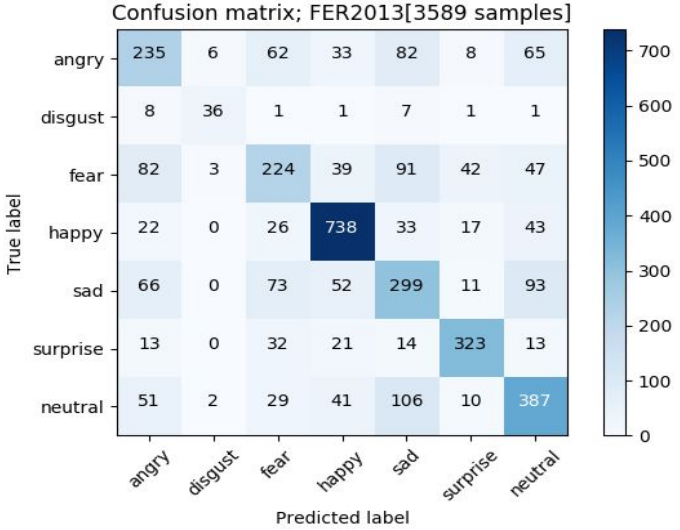
Facial Expression Recognition (FER)

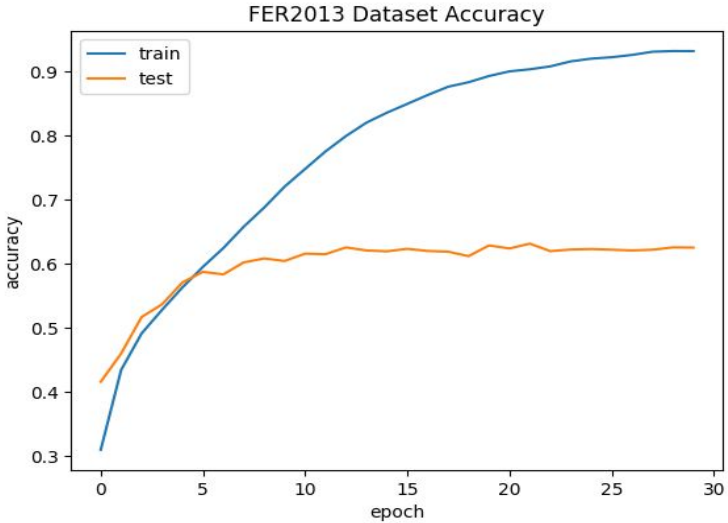
- **MODEL:** FER model is trained using FER-2013 dataset. In test phase 1, captured images of face (resized to 48p x 48p, greyscale) are periodically fed to a CNN (convolutional neural network) model with layers of convolutions followed by max pooling implemented in keras (Python) for classification of facial expression. Emotion classes: angry, disgusted, sad, happy, surprised, neutral, fearful.
- **FRONTEND:** React modules: react-webcam for capturing video feed.
- **BACKEND:** The feed from frontend is sent to Django API which does the processing and returns the percentages of the 7 emotion classes.

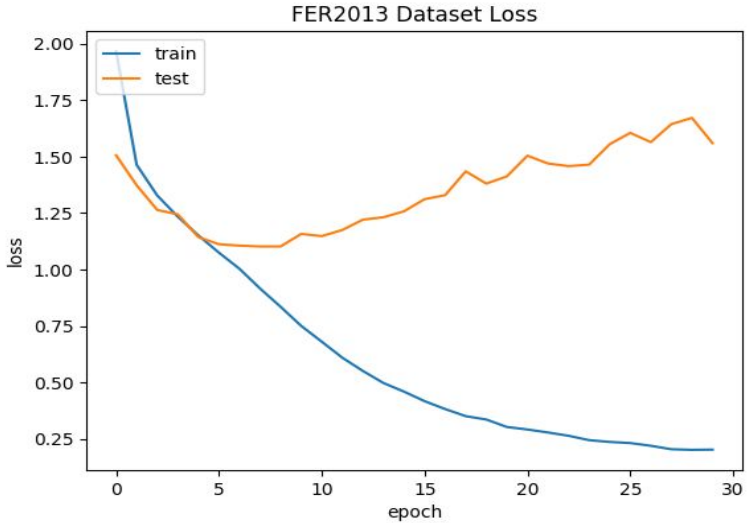


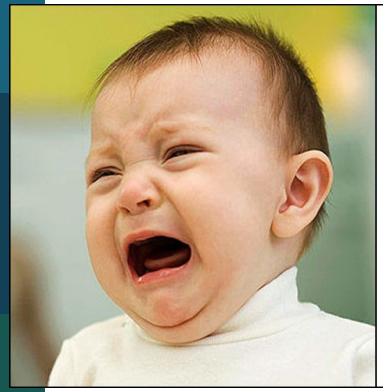

Grayscale Image
48x48
convolution
64 Feature maps
46x46
convolution
64 Feature maps
44x44
max pooling
64 Feature maps
43x43
convolution
128 Feature maps
41x41
convolution
optimizer: Adam
learning rate: 0.0001
batch_size: 256
dense layer
1024 nodes
output layer
(softmax)
128 Feature maps
39x39
convolution
128 Feature maps
37x37
convolution
128 Feature maps
35x35
max pooling
128 Feature maps
34x34
flatten
1-Dimensional
vector

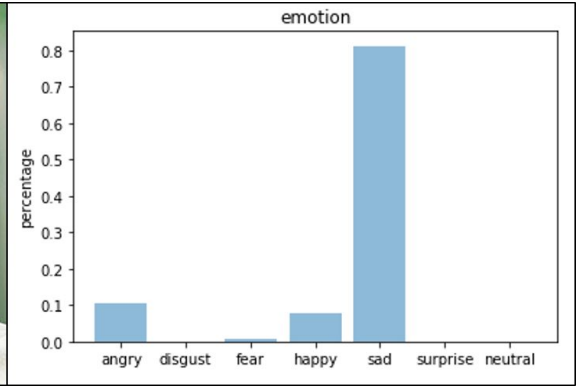
- Dataset: FER-2013 (grayscale images of faces with varied expressions). Source: https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recog nition -challenge/data
- Training Samples: 28,709
- Validation Samples: 3,589
- Validation Accuracy: 63%

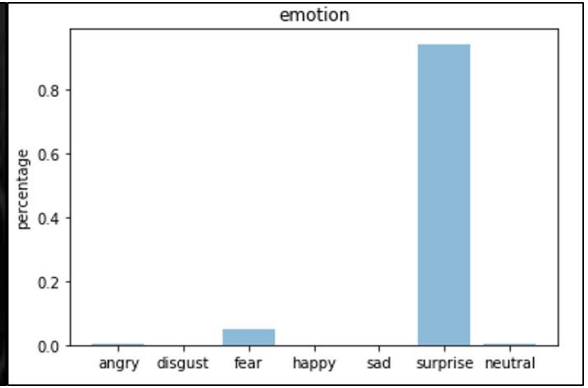


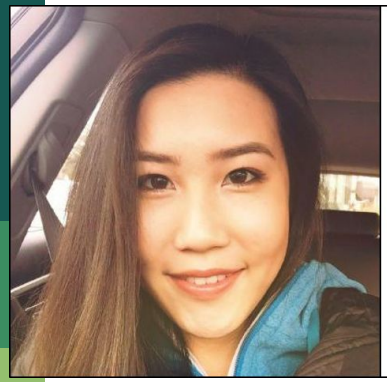


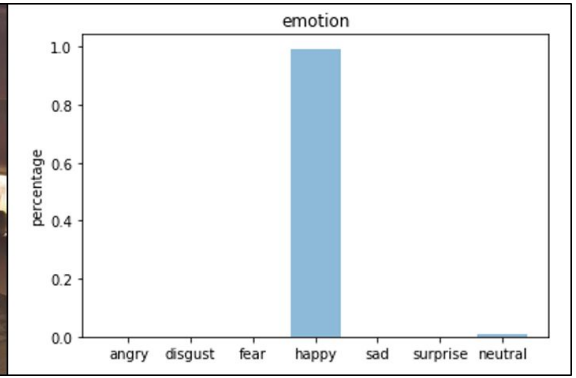


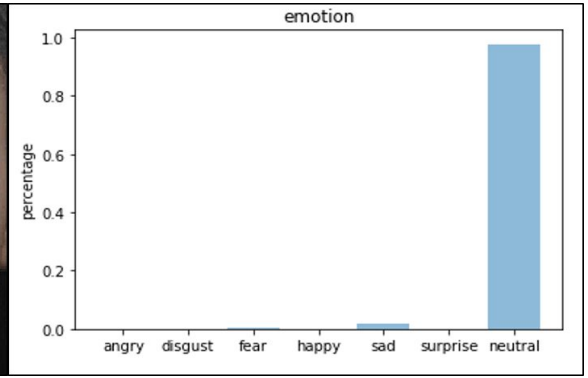








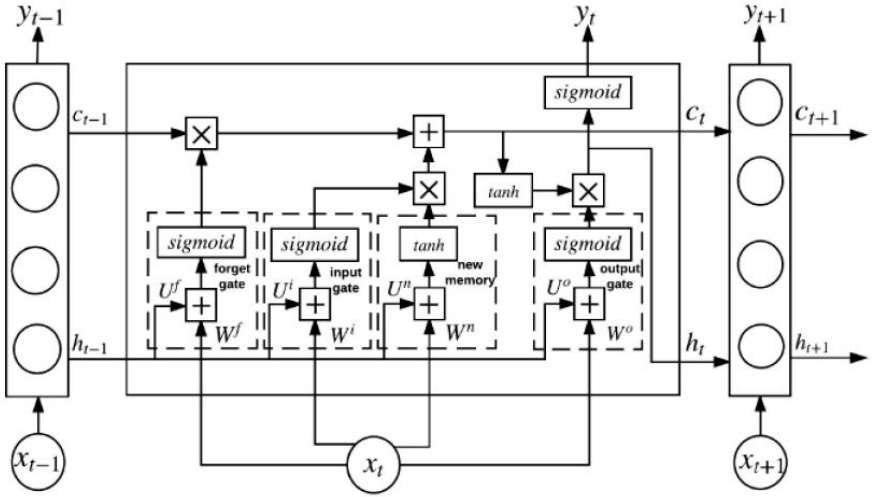




Testing the model on images not in dataset

Speech Emotion Recognition (SER)

- **MODEL:** In test phase 1, the recorded audio (cropped into sequences of 60000) is periodically fed to an LSTM (Long Short Term Memory) model implemented in keras (Python) for classification. Emotion classes: angry, disgusted, sad, happy, surprised, neutral, fearful.
- **FRONTEND:** React modules: react-mic for capturing audio feed.
- **BACKEND:** The feed from frontend is sent to Django API which does the processing and returns the percentages of the 7 emotion classes.



$$f_t = \sigma(W^f x_t + U^f h_{t-1})$$

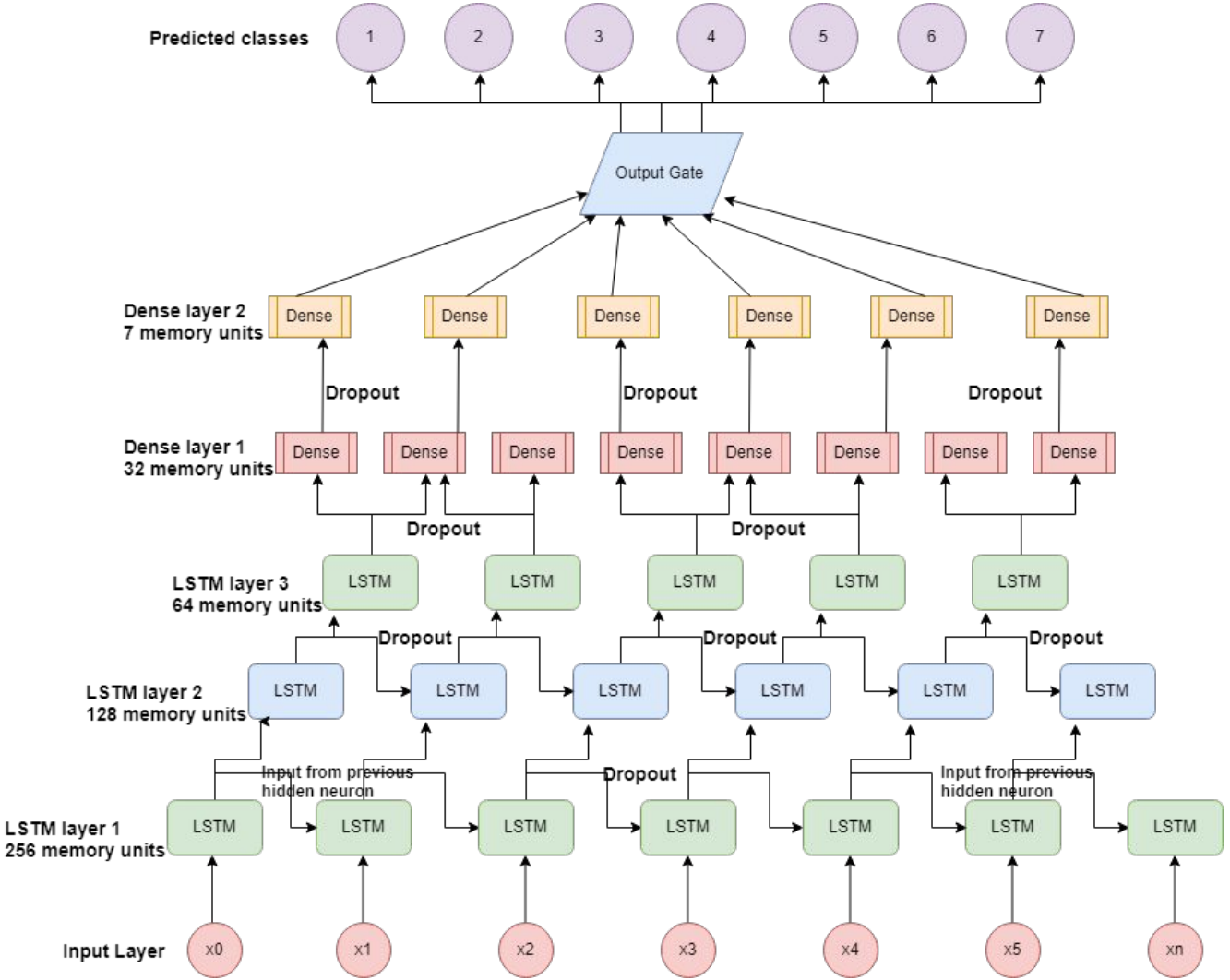
$$i_t = \sigma(W^i x_t + U^i h_{t-1})$$

$$\tilde{C}_t = \tanh(W^n x_t + U^n h_{t-1})$$

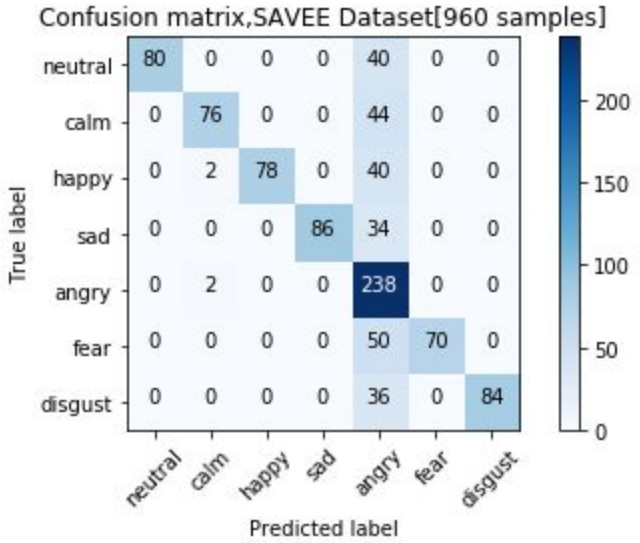
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

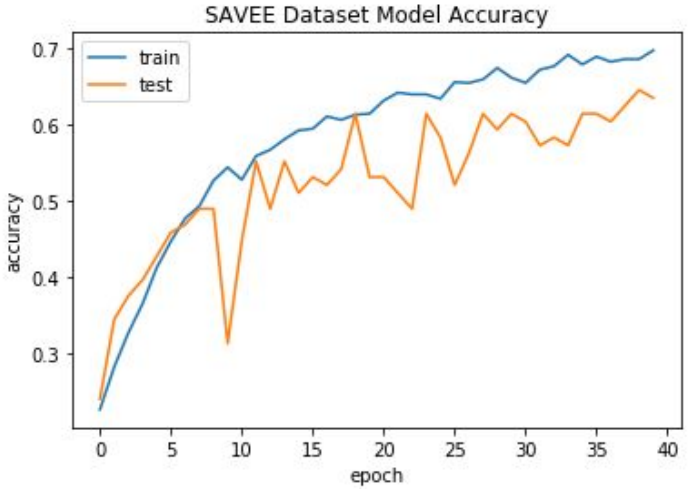
$$o_t = \sigma(W^o x_t + U^o h_{t-1})$$

$$h_t = o_t * \tanh(C_t)$$

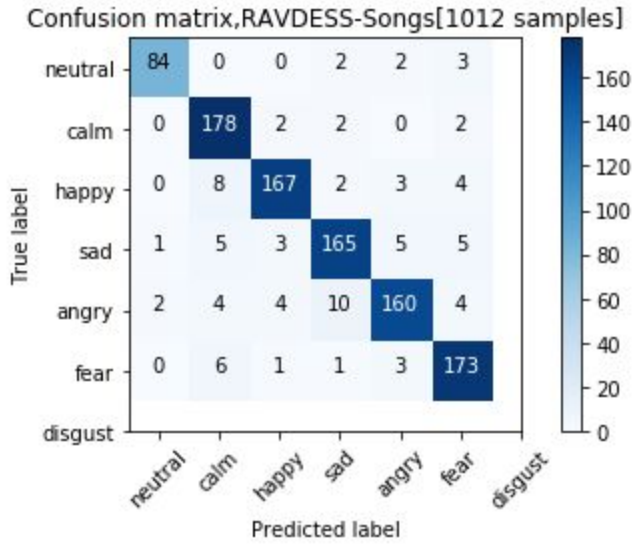

Predicted classes
1
2
3
4
5
6
7
Output Gate
Dense layer 2
7 memory units
Dense
Dropout
Dense layer 1
32 memory units
LSTM layer 3
64 memory units
LSTM
LSTM layer 2
128 memory units
Input from previous
hidden neuron
LSTM layer 1
256 memory units
Input Layer
x0
x1
x2
x3
x4
x5
xn

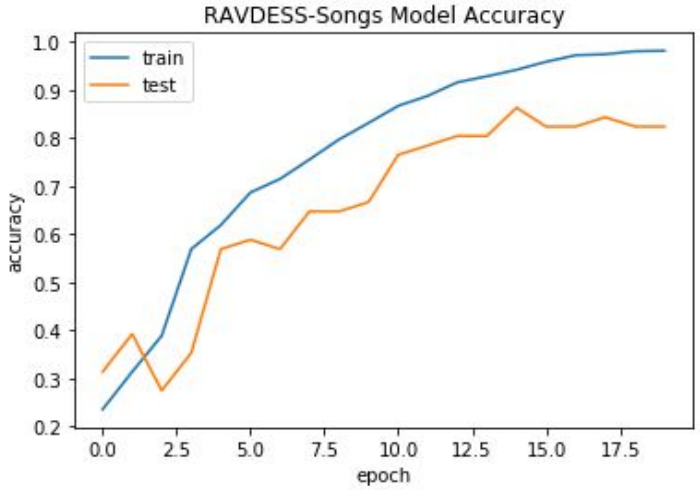
- Dataset: Surrey Audio-Visual Expressed Emotion (SAVEE); 4 actors speaking different sentences in various emotions. Source: http://kahlan.eps.surrey.ac.uk/savee
- Training Samples: 864
- Validation Samples: 96
- Validation Accuracy: 63.54%

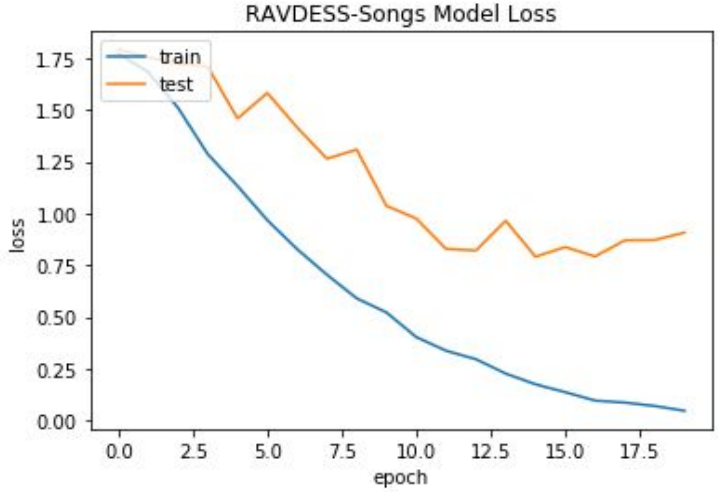




- Dataset: Audio songs subset of Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS). Source: https://smartlaboratory.org/ravdess
- Training Samples: 960
- Validation Samples: 52
- Validation Accuracy: 82.35%







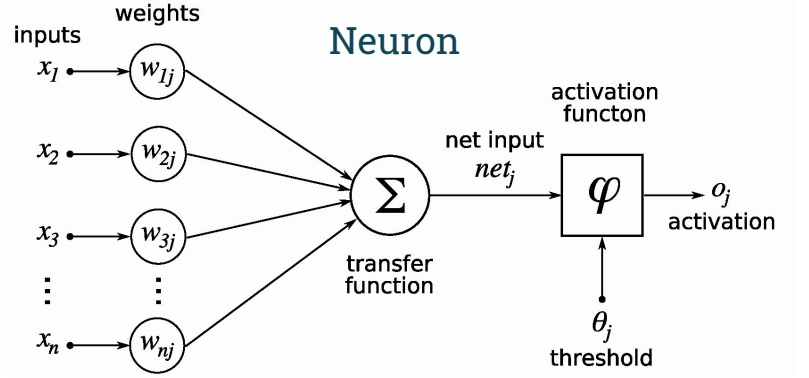
Confidence Score and Certainty Score

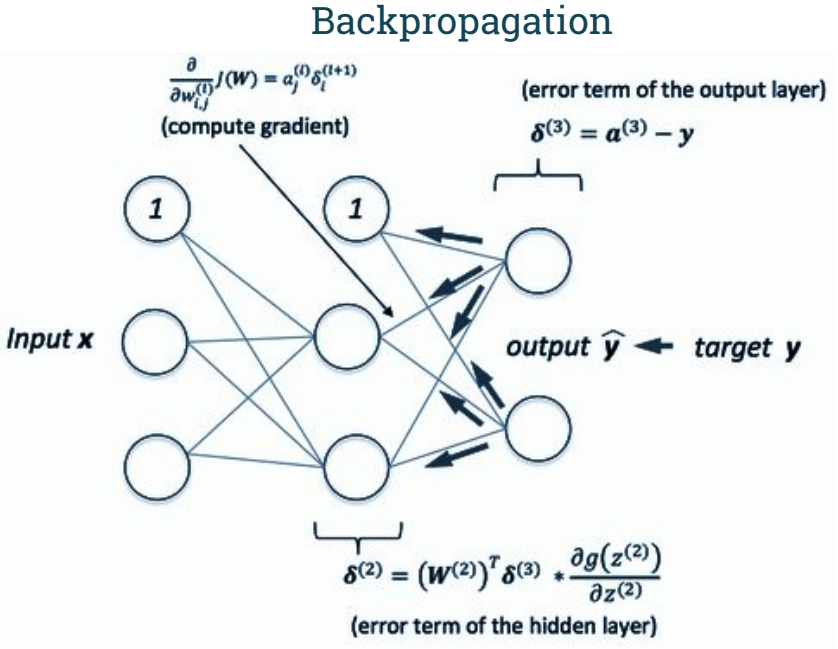
- In test phase 1, speech or verbal response is converted to text using speech recognition library
- Text is preprocessed (conversion to lowercase, expansion of contractions (e.g., I've I have), removal of punctuations and other symbols).
- There are 12 sets containing 1-grams, 2-grams and 3-grams reflecting high confidence, low confidence, high certainty, and low certainty.
- Words that depict confidence levels are usually adjectives, adverbs, nouns and verbs (e.g., assure, conclude, without a doubt, sneaking suspicion, etc.) whereas words that reflect certainty levels are usually modals, helping verbs and adverbs (e.g., might, maybe, definitely, surely, etc.).
- Both confidence and certainty scores lie between -1 and 1, and are weighted according to the value of n in n-gram. Higher the score, higher the confidence or certainty.

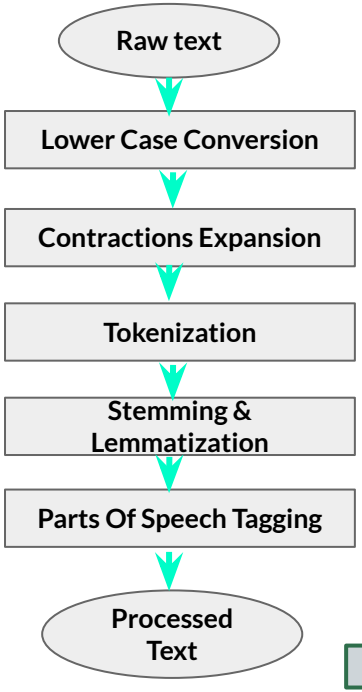
```
preprocess(text)
split_into_ngrams(text)
p1, p2, p3 = cardinality of intersection set of text n-grams
                with confident/ certain n-grams
psum = p1 + 2*p2 + 3*p3
n1, n2, n3 = cardinality of intersection set of text n-grams
                with unconfident/ uncertain n-grams
nsum = n1 + 2*n2 + 3*n3
if (psum+nsum)!=0:
        confidence/certainty_score =
                        (psum-nsum)/(psum+nsum)
else:
        confidence/certainty_score = 0
```

Sentiment Analysis of Image-based Description

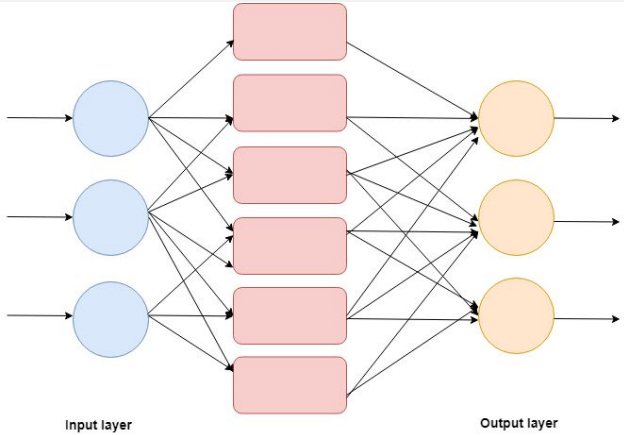
- **MODEL:** Sentiment classification model is implemented in keras (Python) and trained using Amazon product reviews, Stanford movie reviews and Twitter Airline tweets. In test phase 2, the image description is converted to a 3-feature vector and fed to the artificial neural network model.
- **FRONTEND:** The images from the database (Node Server) are fetched from the file system and displayed. A textarea where the description can be written.
- **BACKEND:** The input from frontend is sent to the API (Django Server) for processing and it return the sentiment scores of three classes.



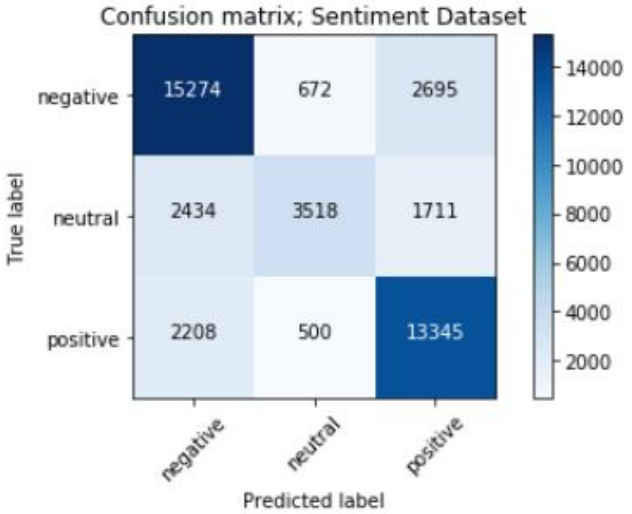


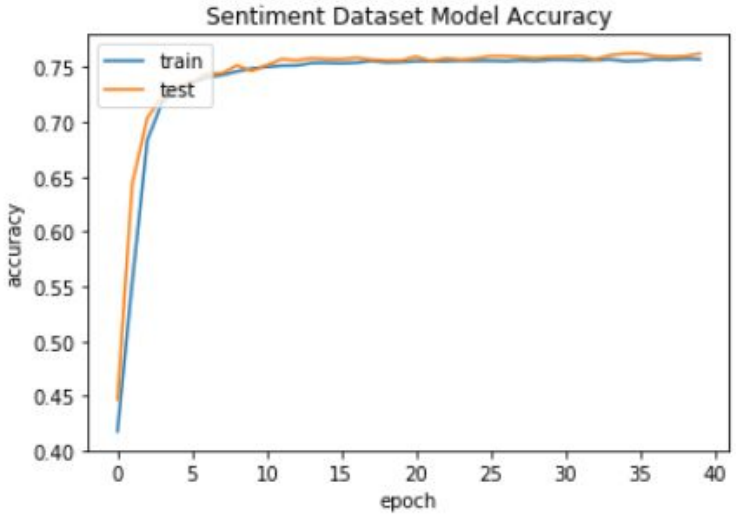


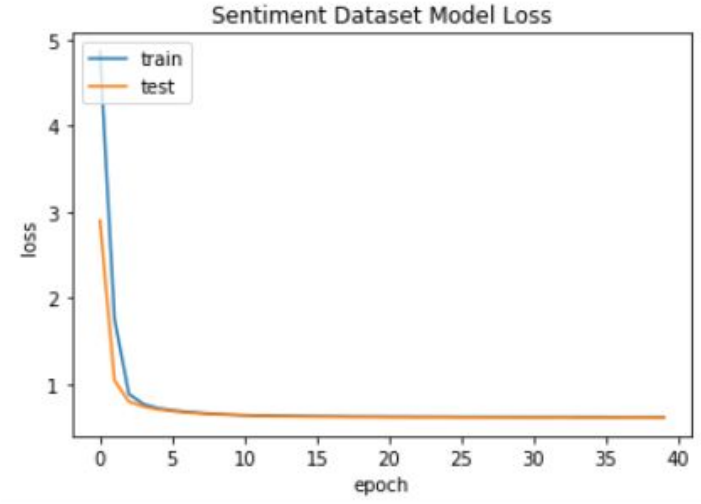

```
for each word w:

if (count of w in class X = 0)
      rv(w, X) = 0
else if (count of w in class X = total count of w in all classes)
      rv(w, X) = 2.5 * count of w in class X
else
      rv(w, X) = 1/log10(total count of w in all classes/ count of
w in class X)
```

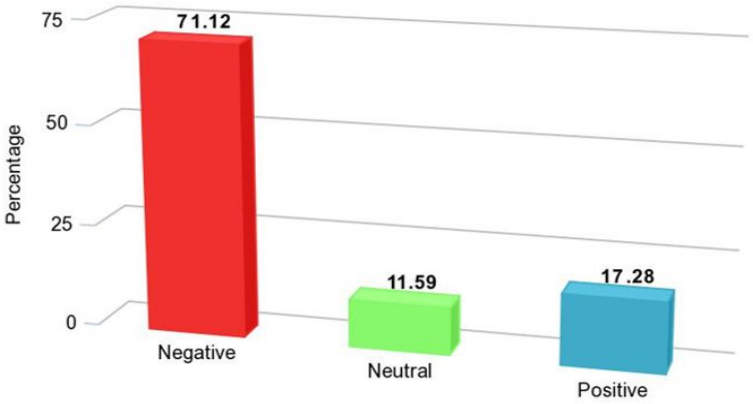


- Dataset: Amazon Reviews, Sentiment Movie Reviews, Twitter Airline Sentiment. Sources:
  - http://jmcauley.ucsd.edu/data/amazon
  - https://nlp.stanford.edu/sentiment/index.html
  - https://www.kaggle.com/crowdflower/twitter-airline-sentiment
- Training Samples: 36, 003
- Validation Samples: 6, 354
- Validation Accuracy: 76.24%



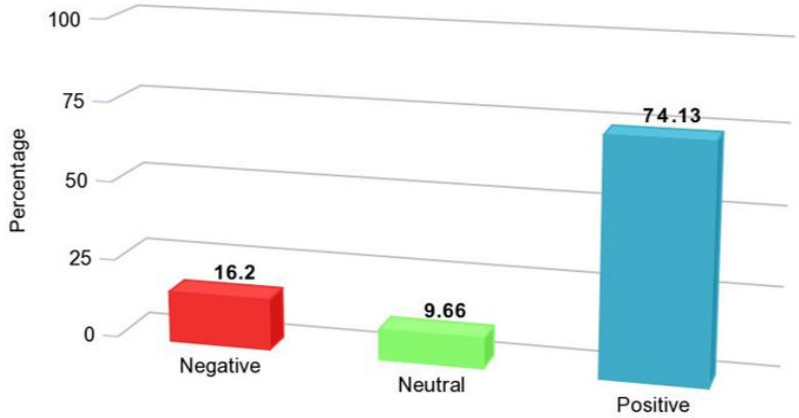






Text example 1
The lion is cowering before the hyenas, tired, weak and afraid.

Testing the model on text not in dataset



Text example 2
A cheerful bird, sitting in the rain and enjoying every minute of it.

Adaptive Interview Chatbot

- **MODEL:**
  - This is a part of test phase 3. Chatbot is trained using general conversation or small talk lists (937 lists) from .yml and .txt files. It later consults sqlite database generated from the lists.
  - Q/A system parses .json file to select questions and match responses.
  - Questions are chosen based on difficulty and user's performance. Answers are scored using cosine and jaccard similarity measures.
  - User can ask domain-specific questions post interview.
- **FRONTEND:** A scrollable chat window
- **BACKEND:** User's response is sent to server and bot's response is fetched and displayed.

Levenshtein distance between two words is the minimum number of edits required to change one string to another.

$$\operatorname{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0, \\ \min \begin{cases} \operatorname{lev}_{a,b}(i-1,j) + 1 \\ \operatorname{lev}_{a,b}(i,j-1) + 1 \\ \operatorname{lev}_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases}$$

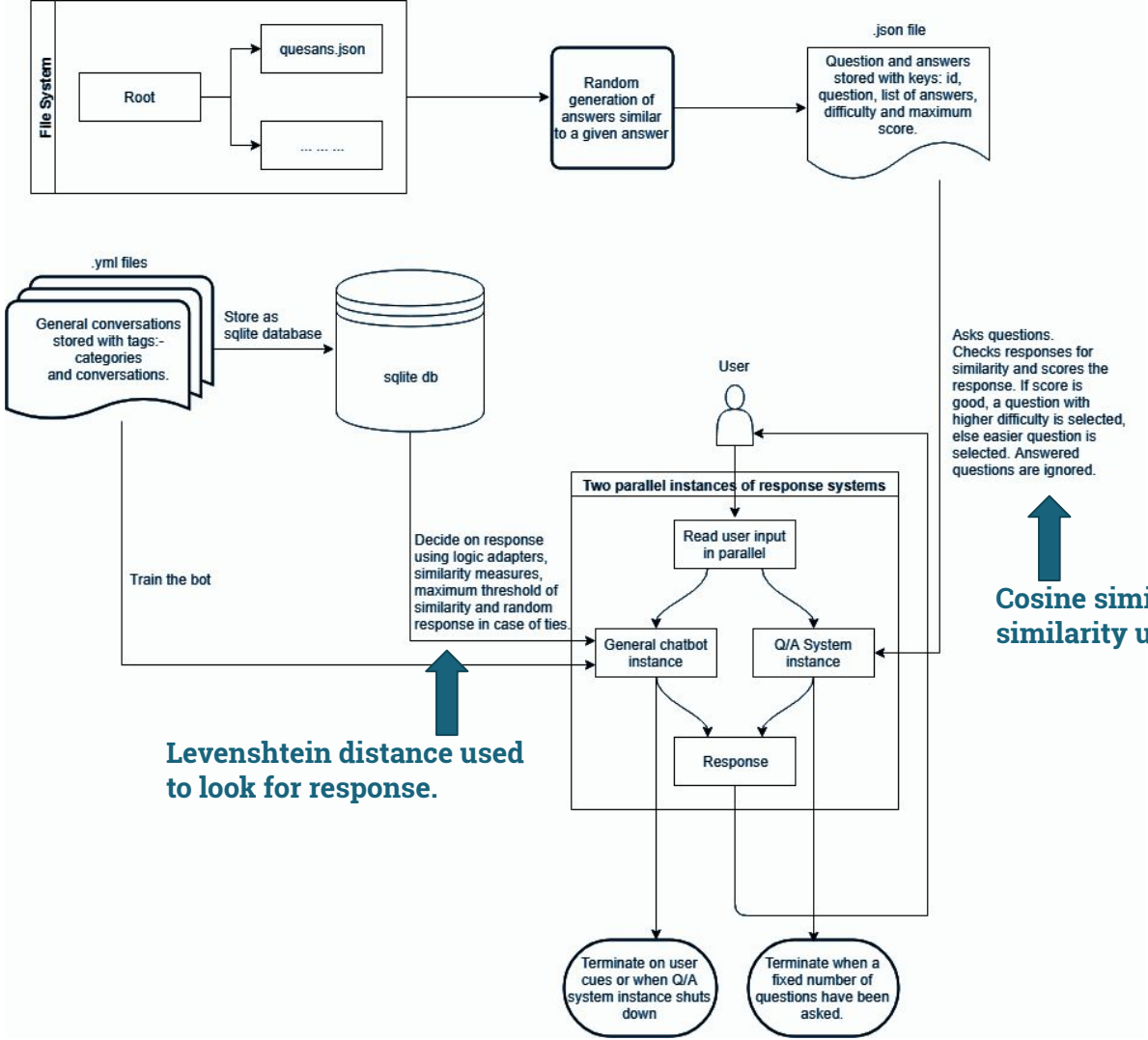
Given and actual responses are count-vectorized to A and B, cosine similarity is the cosine of angle between the two vectors is calculated.

$$similarity = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|_2 \|\mathbf{B}\|_2}$$

$$\frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|_2 \|\mathbf{B}\|_2} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Jaccard Similarity determines the ratio of common words (or n-grams) to total words.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

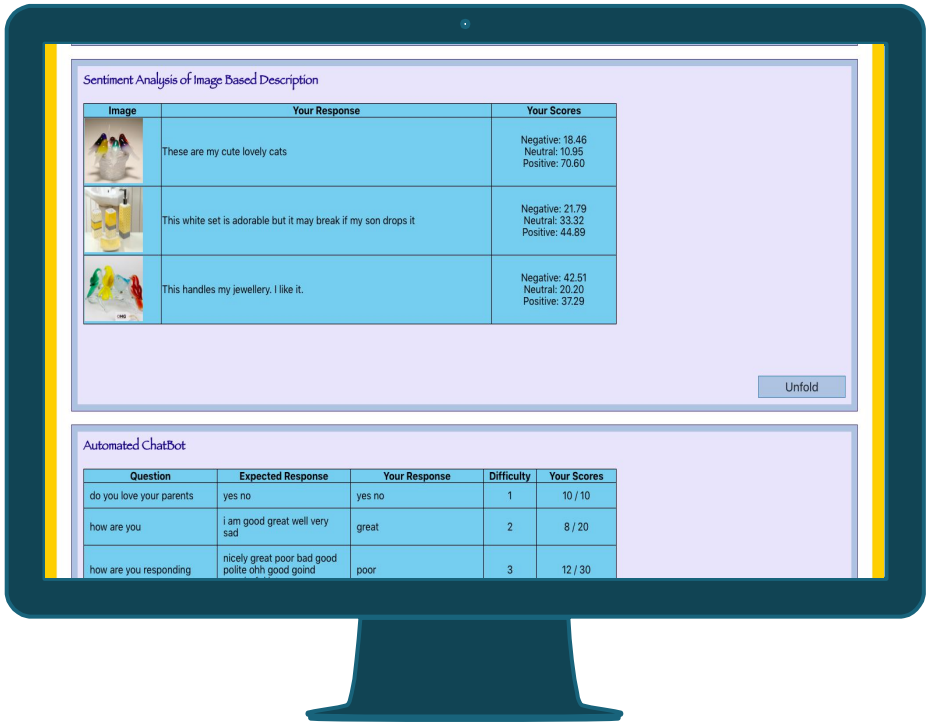

File System
Root
quesans.json
... ...
Random generation of answers similar to a given answer
.json file
Question and answers stored with keys: id, question, list of answers, difficulty and maximum score.
.yml files
General conversations stored with tags:- categories and conversations.
Store as sqlite database
sqlite db
User
Asks questions. Checks responses for similarity and scores the response. If score is good, a question with higher difficulty is selected, else easier question is selected. Answered questions are ignored.
Two parallel instances of response systems
Read user input in parallel
Decide on response using logic adapters, similarity measures, maximum threshold of similarity and random response in case of ties.
Train the bot
General chatbot instance
Q/A System instance
Response
Terminate on user cues or when Q/A system instance shuts down
Terminate when a fixed number of questions have been asked.
Cosine similarity and Jaccard similarity used to score answer.
Levenshtein distance used to look for response.

```
function score_answer(response,
expected_response, difficulty):
    cosine1 = cosine_similarity_1gram(response,
expected_response)
    cosine2 =
cosine_similarity_2gram(response,
expected_response)
    jaccard = jaccard_similarity(response,
expected_response)
    score = (cosine1 * 6) + (jaccard * 2.4)
    if cosine2 < 0.2:          score += cosine2
    else if cosine2 < 0.3:     score += 0.4
    else if cosine2 < 0.5:     score += 0.8
    else if cosine2 < 0.7:     score += 1
    else:                      score += 1.6
    return ceiling(score) * difficulty
```

```
function ask_question(highscore, n, index):
 if n == max_num_of_ques_to_be_asked:
    Select question of median difficulty
 else if n > 0:
    if highscore:
        Choose a question with a difficulty higher than
        that of question marked by index
      if there is no question with higher difficulty:
        Choose a question with a lower difficulty which
        is higher than all unanswered questions of
        lower difficulty
    else:
        Choose a question with a difficulty lower than
        that of question marked by index
      if there is no question with higher difficulty:
        Choose a question with a higher difficulty which
        is lower than all unanswered questions of
        higher difficulty
 else:
    End interview
```

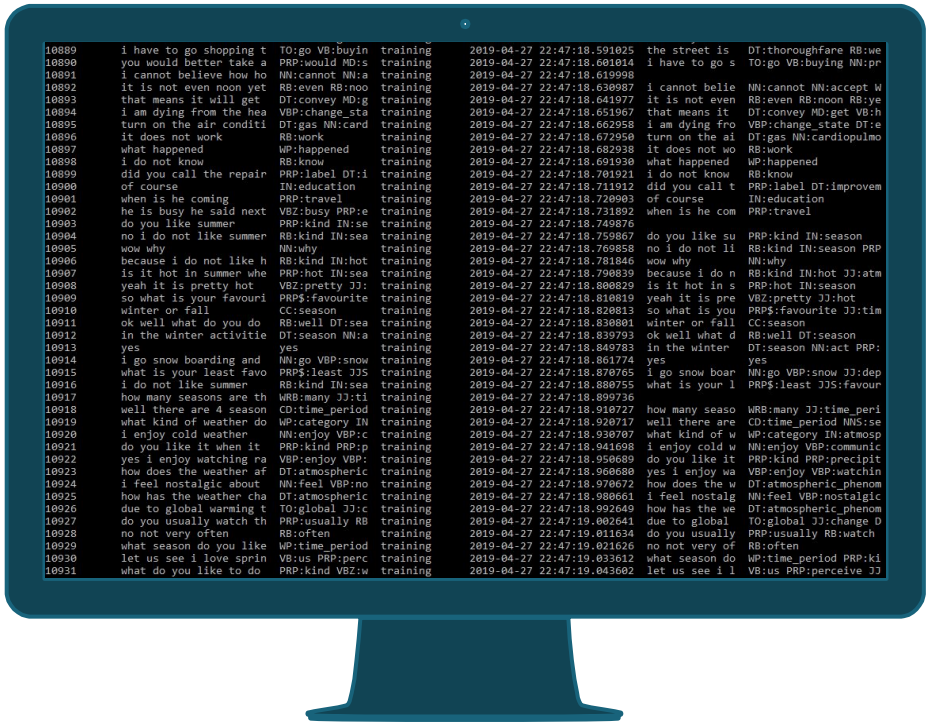
User Interface: Frontend

- The frontend implementation uses ReactJS and NodeJS frameworks. Bootstrap 4 is used for UI. There are different tabs on navigation bar for creating test, taking test and viewing results' report.
- There are three tabs for the three phases of test- FER/SER/confidence/certainty analysis, sentiment analysis and interview with chatbot.
- Three-tier client server architecture is used because data is stored in database (File System) can be accessed and modified by the user with the help of the GUI between them.
- Frontend output is the response to any event or action performed on the website that can be seen in the form of routing to different pages, the change of questions or images on next and previous buttons, starting, submitting and moving through the phases of the test, etc.



User Interface: Backend

- The backend implementation uses NodeJS and Django (Python) frameworks. The file system interaction is handled by the NodeJS, for example, saving the images in folders, question-answers in .json files in separate folders for each user for each test.
- The processing of inputs and their responses in terms of score is generated by the python server and returned to the frontend. All the inputs to the NodeJS server result in either publishing the data to files and folders or fetching it from there.
- The responses of the user in different phases are also saved in the .json files. These responses are sent to python server for evaluation and determination of score by using Machine Learning models or Chatbot system.



10889	i have to go shopping t	TO:go VB:buyin	training	2019-04-27 22:47:18.591025	the street is	DT:thoroughfare RB:we
10890	you would better take a	PRP:would MD:s	training	2019-04-27 22:47:18.601014	i have to go s	TO:go VB:buying NN:pr
10891	i cannot believe how ho	NN:cannot NN:a	training	2019-04-27 22:47:18.619998		
10892	it is not even noon yet	RB:even RB:noo	training	2019-04-27 22:47:18.630987	i cannot belie	NN:cannot NN:accept W
10893	that means it will get	DT:convey MD:g	training	2019-04-27 22:47:18.641977	it is not even	RB:even RB:noon RB:ye
10894	i am dying from the hea	VBP:change_sta	training	2019-04-27 22:47:18.651967	that means it	DT:convey MD:get VB:h
10895	turn on the air conditi	DT:gas NN:card	training	2019-04-27 22:47:18.662958	i am dying fro	VBP:change_state DT:e
10896	it does not work	RB:work	training	2019-04-27 22:47:18.672950	turn on the ai	DT:gas NN:cardiopulmo
10897	what happened	WP:happened	training	2019-04-27 22:47:18.682938	it does not wo	RB:work
10898	i do not know	RB:know	training	2019-04-27 22:47:18.691930	what happened	WP:happened
10899	did you call the repair	PRP:label DT:i	training	2019-04-27 22:47:18.701921	i do not know	RB:know
10900	of course	IN:education	training	2019-04-27 22:47:18.711912	did you call t	PRP:label DT:improvem
10901	when is he coming	PRP:travel	training	2019-04-27 22:47:18.720903	of course	IN:education
10902	he is busy he said next	VBZ:busy PRP:e	training	2019-04-27 22:47:18.731892	when is he com	PRP:travel
10903	do you like summer	PRP:kind IN:se	training	2019-04-27 22:47:18.749876		
10904	no i do not like summer	RB:kind IN:sea	training	2019-04-27 22:47:18.759867	do you like su	PRP:kind IN:season
10905	wow why	NN:why	training	2019-04-27 22:47:18.769858	no i do not li	RB:kind IN:season PRP
10906	because i do not like h	RB:kind IN:hot	training	2019-04-27 22:47:18.781846	wow why	NN:why
10907	is it hot in summer whe	PRP:hot IN:sea	training	2019-04-27 22:47:18.790839	because i do n	RB:kind IN:hot JJ:atm
10908	yeah it is pretty hot	VBZ:pretty JJ:	training	2019-04-27 22:47:18.800829	is it hot in s	PRP:hot IN:season
10909	so what is your favouri	PRP$:favourite	training	2019-04-27 22:47:18.810819	yeah it is pre	VBZ:pretty JJ:hot
10910	winter or fall	CC:season	training	2019-04-27 22:47:18.820813	so what is you	PRP$:favourite JJ:tim
10911	ok well what do you do	RB:well DT:sea	training	2019-04-27 22:47:18.830801	winter or fall	CC:season
10912	in the winter activitie	DT:season NN:a	training	2019-04-27 22:47:18.839793	ok well what d	RB:well DT:season
10913	yes	yes	training	2019-04-27 22:47:18.849783	in the winter	DT:season NN:act PRP:
10914	i go snow boarding and	NN:go VBP:snow	training	2019-04-27 22:47:18.861774	yes	yes
10915	what is your least favo	PRP$:least JJS	training	2019-04-27 22:47:18.870765	i go snow boar	NN:go VBP:snow JJ:dep
10916	i do not like summer	RB:kind IN:sea	training	2019-04-27 22:47:18.880755	what is your l	PRP$:least JJS:favour
10917	how many seasons are th	WRB:many JJ:ti	training	2019-04-27 22:47:18.899736		
10918	well there are 4 season	CD:time_period	training	2019-04-27 22:47:18.910727	how many seaso	WRB:many JJ:time_peri
10919	what kind of weather do	WP:category IN	training	2019-04-27 22:47:18.920717	well there are	CD:time_period NNS:se
10920	i enjoy cold weather	NN:enjoy VBP:c	training	2019-04-27 22:47:18.930707	what kind of w	WP:category IN:atmosp
10921	do you like it when it	PRP:kind PRP:p	training	2019-04-27 22:47:18.941698	i enjoy cold w	NN:enjoy VBP:communic
10922	yes i enjoy watching ra	VBP:enjoy VBP:	training	2019-04-27 22:47:18.950689	do you like it	PRP:kind PRP:precipit
10923	how does the weather af	DT:atmospheric	training	2019-04-27 22:47:18.960680	yes i enjoy wa	VBP:enjoy VBP:watchin
10924	i feel nostalgic about	NN:feel VBP:no	training	2019-04-27 22:47:18.970672	how does the w	DT:atmospheric_phenom
10925	how has the weather cha	DT:atmospheric	training	2019-04-27 22:47:18.980661	i feel nostalg	NN:feel VBP:nostalgic
10926	due to global warming t	TO:global JJ:c	training	2019-04-27 22:47:18.992649	how has the we	DT:atmospheric_phenom
10927	do you usually watch th	PRP:usually RB	training	2019-04-27 22:47:19.002641	due to global	TO:global JJ:change D
10928	no not very often	RB:often	training	2019-04-27 22:47:19.011634	do you usually	PRP:usually RB:watch
10929	what season do you like	WP:time_period	training	2019-04-27 22:47:19.021626	no not very of	RB:often
10930	let us see i love sprin	VB:us PRP:perc	training	2019-04-27 22:47:19.033612	what season do	WP:time_period PRP:ki
10931	what do you like to do	PRP:kind VBZ:w	training	2019-04-27 22:47:19.043602	let us see i l	VB:us PRP:perceive JJ

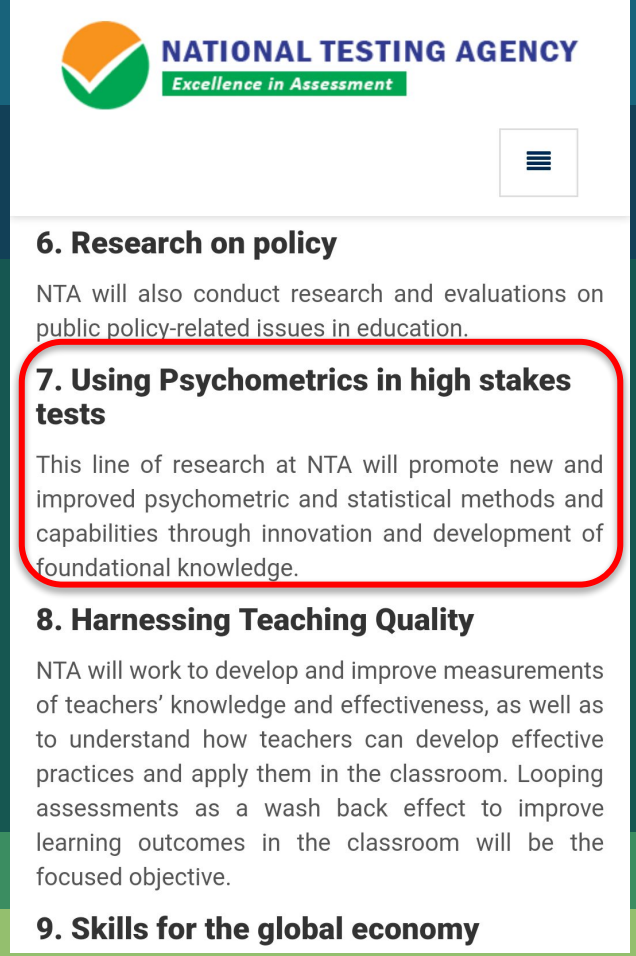
Scope and Future Work

Standardised Testing: There is great scope for research in the field as more and more testing agencies (for example, the NTA) are switching to online statistical and psychometric analysis tools for testing and evaluation for certain jobs.

Bag of Audio Words (BoAW): Speech emotion recognition is a challenging task and the datasets are not available in bigger sizes or numbers. The BoAW representations of acoustic low-level descriptors (LLDs) can been employed for greatly improving the performance on the classification of emotion task.

Addition of more traditional psychometrics tests: More tests can be added based on Item Response Theory (IRT), Rasch Model or Likert Scale for an extensive psychometric analysis. Apart from confidence and certainty score, more text based detections such as persuasive versus dismissive speech, amiable vs unfriendly word choice and inconsistency check can be added.

Flexibility in setting tests and retaining results: Restrictions on the test could be imposed by fixing number and order of questions and images given to the user. Time constraint can be added by fixing the response time per question. Multiple user responses can also be stored for pattern analysis.



Screenshot from the NTA website https://nta.ac.in/Research

References

[1] National Council on Measurement in Education- http://www.ncme.org/ncme/NCME/Resource_Center/Glossary/NCME/Resource_Center/Glossary1.aspx?hkey=4bb87415-44dc-4088-9ed9-e8515326a061#anchorPArchived 2017-07-22 at the Wayback Machine.

[2] Jaswal, Deepika & Vishvanathan, Sowmya & Kp, Soman. (2014). Image Classification Using Convolutional Neural Networks. International Journal of Scientific and Engineering Research. 5. 1661-1668. 10.14299/ijser.2014.06.002.

[3] Schmidhuber, Juergen. (2014). Deep Learning in Neural Networks: An Overview. Neural Networks. 61. 10.1016/j.neunet.2014.09.003.

[4] Hochreiter, Sepp & Schmidhuber, Jürgen. (1997). Long Short-term Memory. Neural computation. 9. 1735-80. 10.1162/neco.1997.9.8.1735.

[5] Cen, Ling & Dong, Minghui & Li, Haizhou & Chan, Paul. (2010). Machine Learning Methods in the Application of Speech Emotion Recognition. 10.5772/8613.

[6] Desai, Mitali & Mehta, Mayuri. (2016). Techniques for sentiment analysis of Twitter data: A comprehensive survey. 149-154. 10.1109/CCAA.2016.7813707.

[7] Khuat, Tung & Duc Hung, Nguyen & Thi My Hanh, Le. (2015). A Comparison of Algorithms used to measure the Similarity between two documents. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET). 4. 1117-1121.

[8] Computer Engineering and Applications Vol. 5, No. 1, February 2016. Comparison Jaccard similarity, Cosine Similarity and Combined Both of the Data Clustering With Shared Nearest Neighbor Method Lisna Zahrotun.

[9] https://keras.io/getting-started/sequential-model-guide

[10] https://keras.io/layers/recurrent

[11] P. Kataria, K. Rode, A. Jain, P. Dwivedi, S. Bhingarkar. User Adaptive Chatbot for Mitigating Depression. International Journal of Pure and Applied Mathematics. Volume 118 No. 16 2018, 349-361.

[12] Schmitt, M., Ringeval, F., & Schuller, B.W. (2016). At the Border of Acoustics and Linguistics: Bag-of-Audio-Words for the Recognition of Emotions in Speech. INTERSPEECH.

[13] https://en.wikipedia.org/wiki/Levenshtein_distance

[14] https://chatterbot.readthedocs.io/en/stable

THANK YOU!