

PROFESSIONAL TRAINING REPORT
at
Sathyabama Institute of Science and Technology
(DEEMED TO BE UNIVERSITY)

Submitted in partial fulfilment of the requirements for the
award of Bachelor of Engineering Degree in Computer
Science and Engineering

By
Prakhar Sahu
(Reg. No. 38110417)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING
SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMILNADU.

MARCH 2022



**SATHYABAMA INSTITUTE
OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

(Established under Section 3 of UGC Act, 1956)

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600119

www.sathyabamauniversity.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Professional Training Report is the bonafide work of **Prakhar Sahu (Reg. No.381104170)** who underwent the professional training in **“EMOTION BASED MOVIE RECOMMENDER SYSTEM USING CNN”** under our supervision from Jan 2022 to Mar 2022.

Internal Guide

DR. A. CHRISTY, M.C.A, Ph.D.

Head of the Department

DR. LAKSHMANAN L, M.E., Ph.D.

DR. S. VIGNESHWARI, M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, **Prakhar Sahu**(Reg. No. 38110417) hereby declare that the Professional Training Report on “**EMOTION BASED MOVIE RECOMMENDER SYSTEM USING CNN**” done by me under the guidance of **DR. A. CHRISTY, MCA, Ph.D.**, at Sathyabama Institute of Science and Technology is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE:



SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. SASIKALA M.E., Ph. D, Dean**, School of Computing, **Dr. S. VIGNESHWARI, M.E., Ph.D., and Dr. Lakshmanan L,M.E., PH.D., Heads of the Department of Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive views.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. A. CHRISTY, MCA., Ph.D.,** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-Teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of my project.

ABSTRACT

In terms of e-commerce, recommender framework for the most part guides customers in a customized manner towards interesting / customized manner towards products in a large space of feasible options. To provide reliable recommendation, the recommender systems need to capture the exact customer's need and preference into a user profile. However, for complexes products/ services such as movies, music, news, user emotions play surprisingly critical roles in the decision making process. The traditional method of user profile system doesn't consider the impact of user emotion, the recommender systems cannot understand and capture the continually developing inclinations of a user. The Movie recommendation is based on notions with regards to a client ' s emotions and inclinations. This project additionally examines the System architecture and its implementation, as well as its evaluation procedure. We believe that our system provides improved recommendation to users because it enables the users to understand the relation between their emotional states and the recommended movies.

TABLE OF CONTENTS

Chapter No.	Title	Page No.
	Abstract	vi
	List of Figures	viii
	List of Tables	ix
1.	INTRODUCTION	
	1.1 Outline of the Project	1
2.	LITERATURE SURVEY	2
3.	Aim And Scope of The Present Investigation	
	3.1 Purpose of the Project	4
	3.2 Project Architecture	5
4.	Experimental Or Materials and Methods Algorithms Used	
	4.1 Introduction	6
	4.2 Working Explanation	7
	4.2.1 Notations	7
	4.2.2 Dataset	7
	4.2.3 Convolutional Neural Networks	8
	4.2.4 Support Vector Machine	10
	4.2.5 Movie Recommender System	11
5.	Results And Discussion, Performance Analysis	13
6.	CONCLUSION and SUMMARY	16
7.	APPENDIX	
	A. Source Code	17
	B. Output	27
8	REFERENCES	28

LIST OF FIGURES

Figure No.	Title	Page No.
3.1	Architecture of the model	5
3.2	Architecture of the model	5
4.1	FER-2013	8
4.2	Architecture of traditional Neural Network	9
4.3	Convolutional Layer	9
4.4	Fully Connected	10
4.5	Linearly Separable Data Points	11
4.6	Best Hyper Plane	11
4.7	Movie Recommending Strategy	12
5.1	Accuracy and Loss	13
5.2	SVM vs CNN	13
5.3	Test Image 1	14
5.4	Test Image 2	14
5.5	Test Image 3	15
5.6	Test Image 4	15
6.1	Output	27

CHAPTER 1

INTRODUCTION

1.1 OUTLINE OF THE PROJECT

A Convolutional Neural Network (ConvNet/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. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

Image recognition is one of the tasks in deep neural networks. Neural networks are computing systems designed to recognize patterns. Their architecture is inspired by the human brain structure, hence the name. They consist of three types of layers: input, hidden layers, and output. The input layer receives a signal, the hidden layer processes it, and the output layer makes a decision or a forecast about the input data. Each network layer consists of interconnected *nodes* (*artificial neurons*) that do the computation.

Emotion recognition is the process of identifying human emotion. People vary widely in their accuracy at recognizing the emotions of others. Use of technology to help people with emotion recognition is a relatively nascent research area. Generally, the technology works best if it uses multiple modalities in context. To date, the most work has been conducted on automating the recognition of facial expressions from video, spoken expressions from audio, written expressions from text, and physiology as measured by wearables.

CHAPTER 2

LITERATURE SURVEY

“Recommender systems could be any framework which produce personalized suggestions as result or guides clients in a customized way useful products in a option space” [Burke 2000]. “The recommender systems are used by business in E-commerce for helping the client in their decision-making by recommending items and providing information” [Schafer et al. 2001]. The three main components of recommender systems are:

- The information of the product;
- The information that user should communicate with the system at the start of the recommending process; and
- An algorithm which combines the information of product and user together to generate a result.

There are five different recommendation techniques, but for advanced and subjective domains like film, music and news, the foremost appropriate techniques square measure cooperative, content-based, knowledge-based or the mixture of those techniques. These systems will establish cross-genre niche while not domain data.

The system makes suggestions by laying out the things that are liked by the client and the information of products to be suggested. The recommendations may be created even though the system has received a variety of ratings. However, content-based systems square measure restricted by the options that square measure expressly related to the objects that they suggest [Burke 2002]. For example, content based suggestion for film could be founded on composed materials of a few film: actors' names, plot summaries, etc.

Moreover, it is impossible to consider all features associated to subjective and complex products such as movie, music or news. An example of content-based filtering system is Reel.com's Movie Matches (www.reel.com). Knowledge-based recommendation systems suggest products based on inferences about a user's needs and preferences. These systems have no start-up problems and do not require user ratings. However, knowledge acquisition is very difficult, for example the system PickAFlick [Burke 2000].

Hybrid recommendation systems mix several recommendation methods to realize better execution with lesser downsides. The user profile not solely contains the general ratings of films, but also includes desired features of the movie. The system presumes that a user's perspective toward a selected feature would possibly vary with the amount of the feature in question. as an example, a user might need a high opinion of slapstick comedy in tiny amounts however not in huge amounts.

CHAPTER 3

AIM AND SCOPE OF THE PRESENT INVESTIGATION

3.1 PURPOSE OF THE PROJECT

The main goal of this project is to build a neural network model which gives an accuracy more than what we have received in previous works as mentioned in the literature review on the same dataset (i.e., FER2013) and be able to work dynamically for movie recommendation. The dataset chosen for the project is FER2013. The pre-trained models from transfer learning will be used to achieve the target. E.g.: Resnet50, MobileNetV2 etc. The model should be able to predict every emotion with similar accuracy and none of the emotions would be excluded. Here we have tried to build a model that should be able to do this on a real-time basis using a webcam or a camera and the time taken to generate the results be minimized as much as possible to facilitate real-life usage.

Disadvantages:

A few types of images the model tends to do poorly on include:

- The human in an unusual position
- Face appears against a background of a similar colour
- Brightness of the picture
- Scale variation (face is very large or small in image)

Advantages:

The model has higher prediction than the basic machine learning model. This is simple algorithm built using simple and open-source packages like TensorFlow and SciPy library.

3.2 PROJECT ARCHITECTURE:

ARCHITECTURE OF THE MODEL:

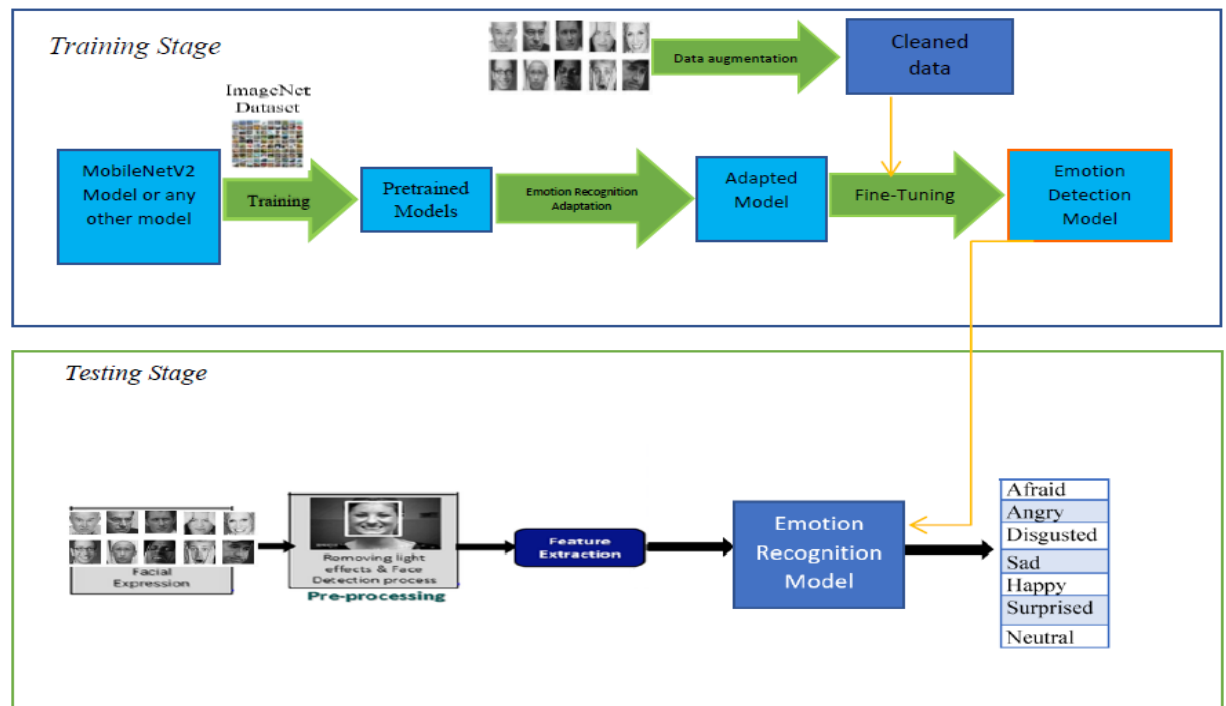


Figure 3. 1

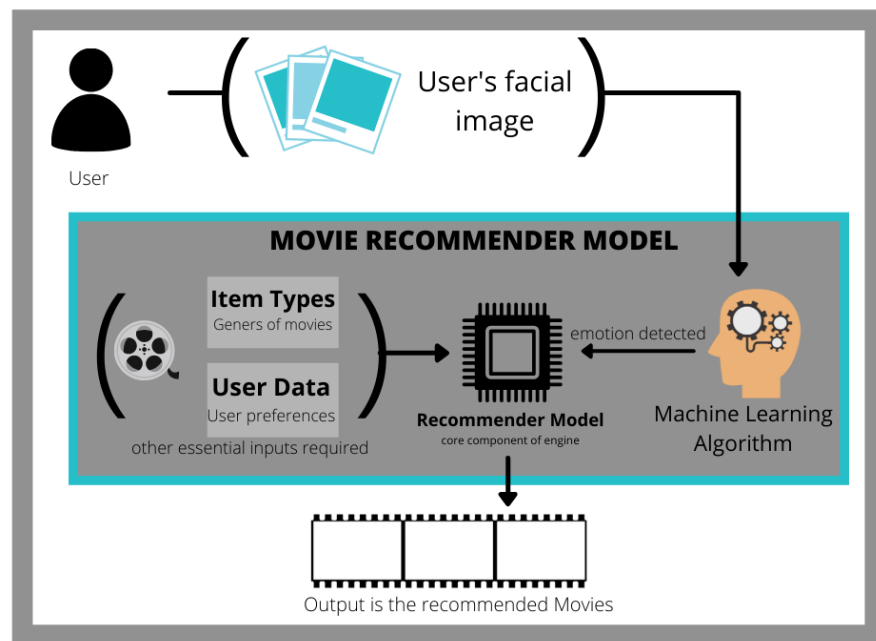


Figure 3.2

CHAPTER 4

EXPERIMENTAL OR MATERIALS AND METHODS

ALGORITHMS USED

4.1 INTRODUCTION

Image recognition is one of the tasks in deep neural networks. Neural networks are computing systems designed to recognize patterns. Their architecture is inspired by the human brain structure, hence the name. They consist of three types of layers: input, hidden layers, and output. The input layer receives a signal, the hidden layer processes it, and the output layer makes a decision or a forecast about the input data. Each network layer consists of interconnected *nodes* (*artificial neurons*) that do the computation.

Traditional neural networks have up to three hidden layers, deep networks may contain hundreds of them. So, to be able to recognize faces, a system must learn their features first. It must be trained to predict whether an object is X or Z. Deep learning models learn these characteristics in a different way from machine learning models. That's why model training approaches are different as well. Each layer of nodes trains on the output (feature set) produced by the previous layer. So, nodes in each successive layer can recognize more complex, detailed features visual representations of what the image depicts. Such a "hierarchy of increasing complexity and abstraction" is known as *feature hierarchy*.

So, the more layers the network has, the greater its predictive capability.

4.2 WORKING EXPLANATION:

4.2.1 Notations:

$$(x, y) = \{(X^{(1)}, y^{(1)}), (X^{(2)}, y^{(2)}), (X^{(3)}, y^{(3)}), \dots, (X^{(m)}, y^{(m)})\}$$

Where $X \in R^{n_x}$, $y \in \{0, 1\}$

$$X = (\uparrow \uparrow \uparrow X^{(1)} X^{(2)} \dots X^{(m)} \downarrow \downarrow \downarrow)$$

Here, $(X(1), X(2), \dots, X(m))$ are columns

$$y = [y^{(1)}, y^{(2)}, \dots, y^{(m)}]$$

$$W = (\uparrow \uparrow \uparrow W^{(1)} W^{(2)} \dots W^{(m)} \downarrow \downarrow \downarrow)^T$$

Where W is considered as the weight matrix usually initialized to 0.

$$A^{[1]} = (\uparrow \uparrow \uparrow a^{1} a^{[1](2)} \dots X^{[1](m)} \downarrow \downarrow \downarrow)$$

Where A represents the Hidden Layer number, [i] represents the ith hidden layer number, and (i) represents the ith element number.

$$Z^{[1]} = (\uparrow \uparrow \uparrow z^{1} z^{[1](2)} \dots z^{[1](m)} \downarrow \downarrow \downarrow)$$

Where Z matrix contains the finally calculated result of [i]th layer.

Dimensions:

Dimension of X: $m \times n_x$

4.2.2 Dataset

The data consists of 48x48 pixel grayscale images of faces. The faces have been automatically registered so that the face is more or less centred and occupies about the same amount of space in each image.

The task is to categorize each face based on the emotion shown in the facial expression into one of seven categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral). The training set consists of 28,709 examples and the public test set consists of 3,589 examples.

Created in 2013, FER-2013 dataset contains 35,887 grayscale 48x48-pixel images, with the images being stored in a spreadsheet where each image's pixel values are stored in cells per row. The images were obtained utilizing Google search and are later grouped per emotion classes, which are anger, disgust, fear,

happiness, neutral, sadness, and surprise. As the dataset was built utilizing Google search, the images are in in-the-wild condition, with even very few images being animated characters.

The dataset originally had data distribution of 28,709 images for training and 3,589 images for public test, but after the competition ended, another 3,589 images which were used for private test were added to the dataset. The usages of FER-2013's data distribution vary among published researches, with each using public test images for different purposes, either as part of training set, validation set, or test set.

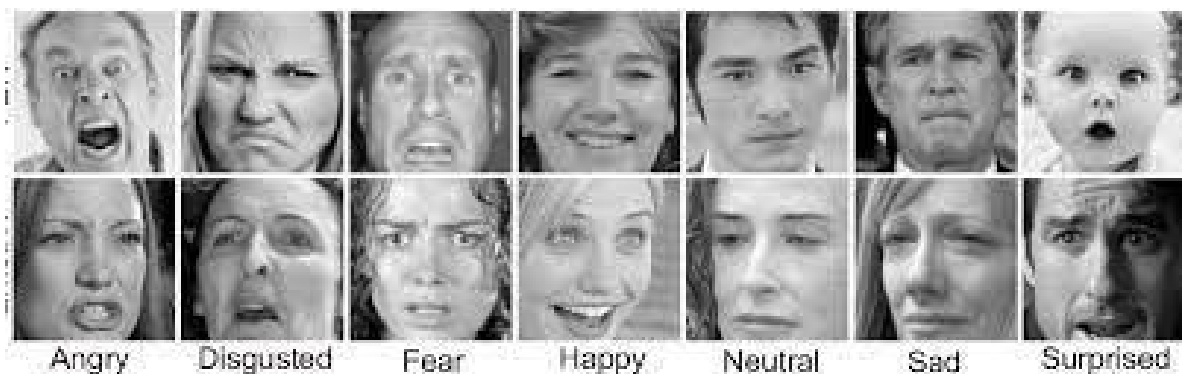


Figure 4. 1 FER-2013

4.2.3 Convolutional Neural Networks

Architecture of a traditional CNN: Convolutional neural networks, also known as CNNs, are a specific type of neural networks that are generally composed of the following layers:

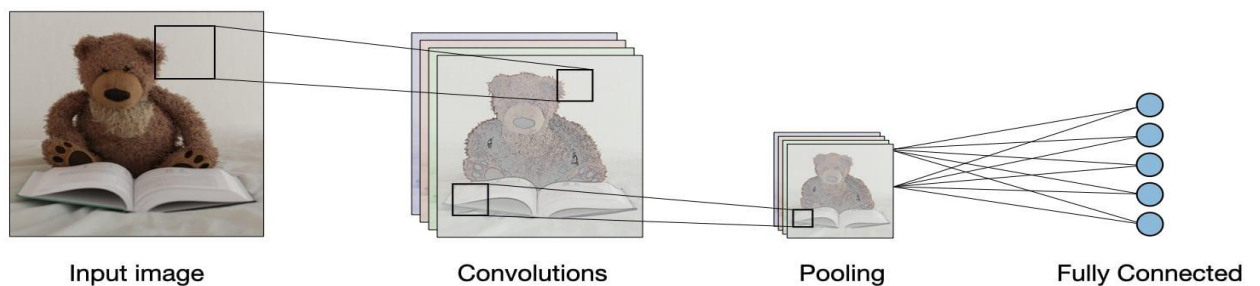


Figure 4. 2 Architecture of traditional Neural Networks

Types of layers:

Convolution layer (CONV) The convolution layer (CONV) uses filters that perform convolution operations as it is scanning the input I with respect to its dimensions. Its hyperparameters include the filter size FF and stride SS . The resulting output OO is called *feature map* or *activation map*.

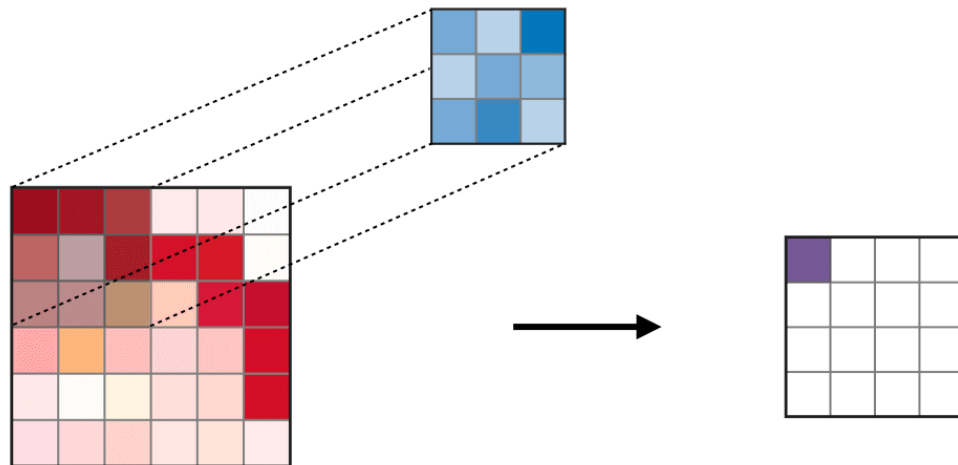
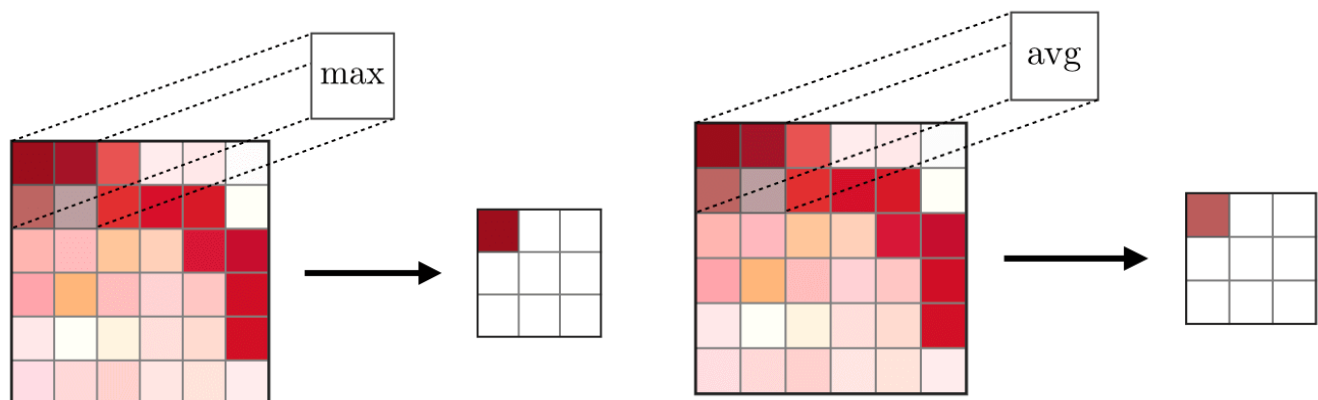


Figure 4. 3 Convolutional Layer

Pooling (POOL) The pooling layer (POOL) is a down sampling operation, typically applied after a convolution layer, which does some spatial invariance. In particular, max and average pooling are special kinds of pooling where the maximum and average value is taken, respectively.



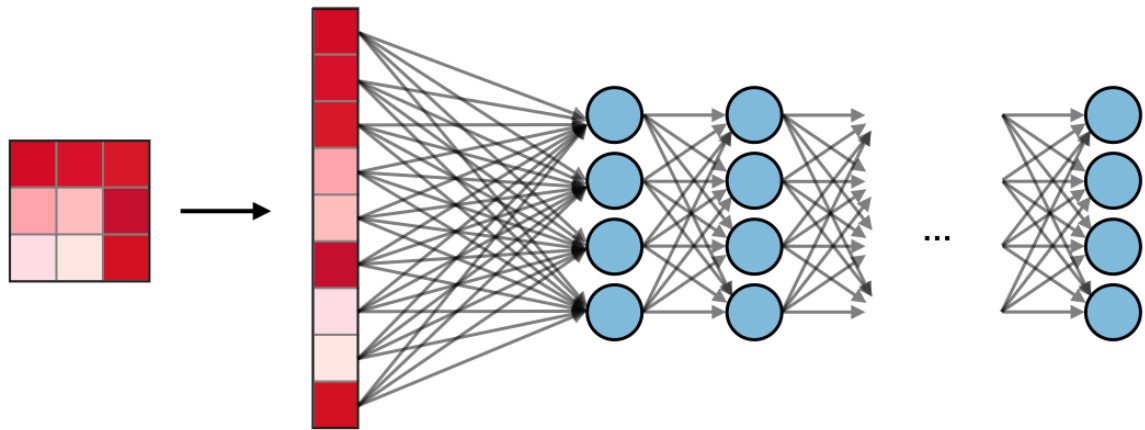


Figure 4. 4 Fully Connected Layer

Fully Connected (FC) The fully connected layer (FC) operates on a flattened input where each input is connected to all neurons. If present, FC layers are usually found towards the end of CNN architectures and can be used to optimize objectives such as class scores.

4.2.4 Support Vector Machine

Support Vector Machine (SVM) is a supervised machine learning algorithm used for both classification and regression. Though we say regression problems as well its best suited for classification. The objective of SVM algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points. The dimension of the hyperplane depends upon the number of features. If the number of input features is two, then the hyperplane is just a line. If the number of input features is three, then the hyperplane becomes a 2-D plane. It becomes difficult to imagine when the number of features exceeds three.

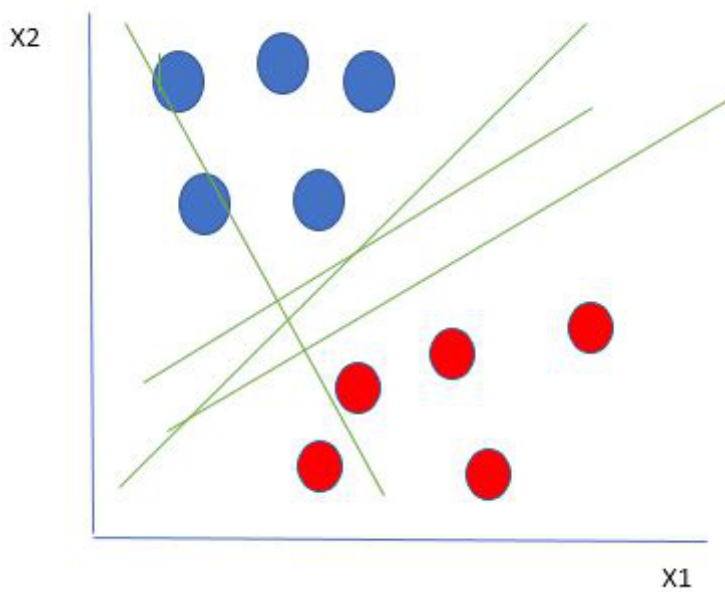


Figure 4. 5 Linearly Separable Data points

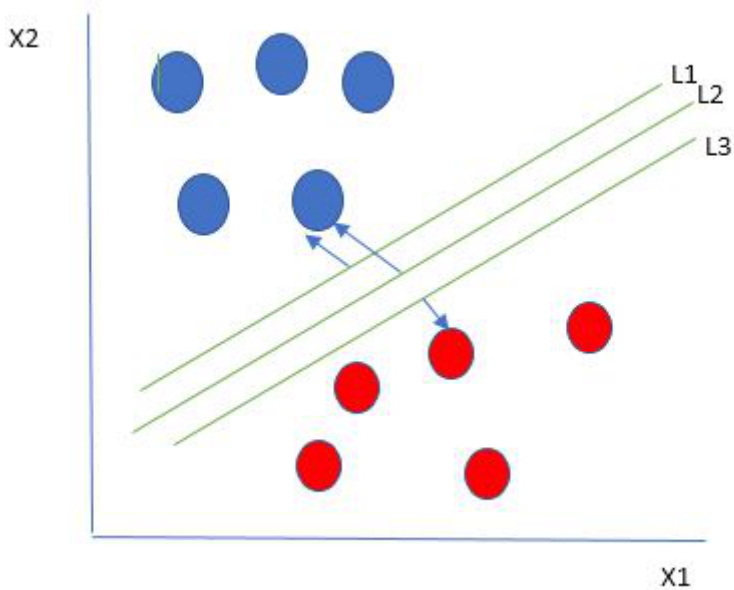


Figure 4. 6 Best Hyper Plane

4.2.5 Movie Recommender System

Recommender System is a system that seeks to predict or filter preferences according to the user's choices. Recommender systems are utilized in a variety of

areas including movies, music, news, books, research articles, search queries, social tags, and products in general. Recommender systems produce a list of recommendations in any of the two ways –

Collaborative filtering: Collaborative filtering approaches build a model from the user's past behaviour (i.e. items purchased or searched by the user) as well as similar decisions made by other users.

Content-based filtering: Content-based filtering approaches uses a series of discrete characteristics of an item in order to recommend additional items with similar properties. It recommends items based on the user's past preferences.

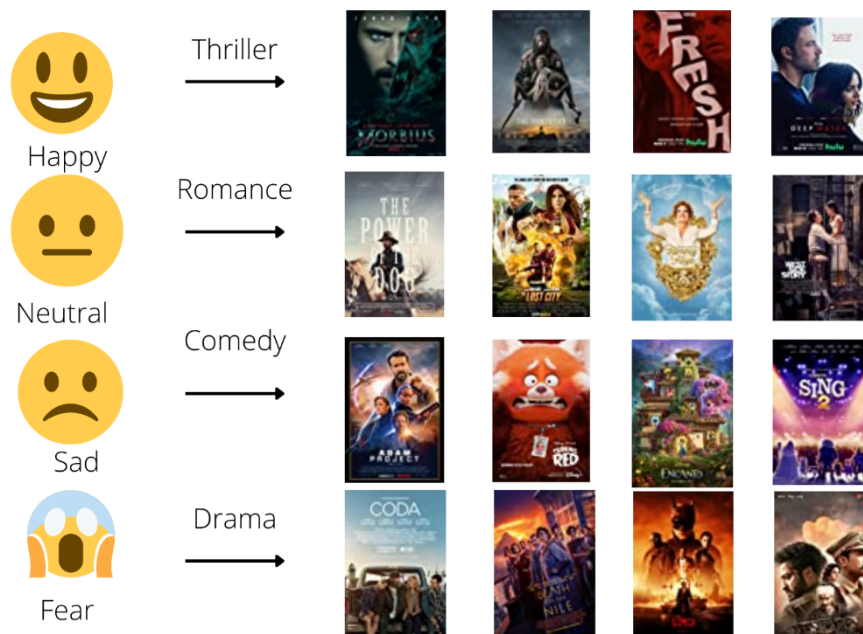


Figure 4.7 Movie Recommending Strategy

CHAPTER 5

RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS

MODEL: LOSS AND ACCURACY PLOT

```
plt.plot(hist.history['val_loss'])  
plt.title('model Loss')  
plt.xlabel('Epochs')  
plt.ylabel('Loss')  
plt.legend(['train', 'test'], loc='upper left')  
plt.show()
```

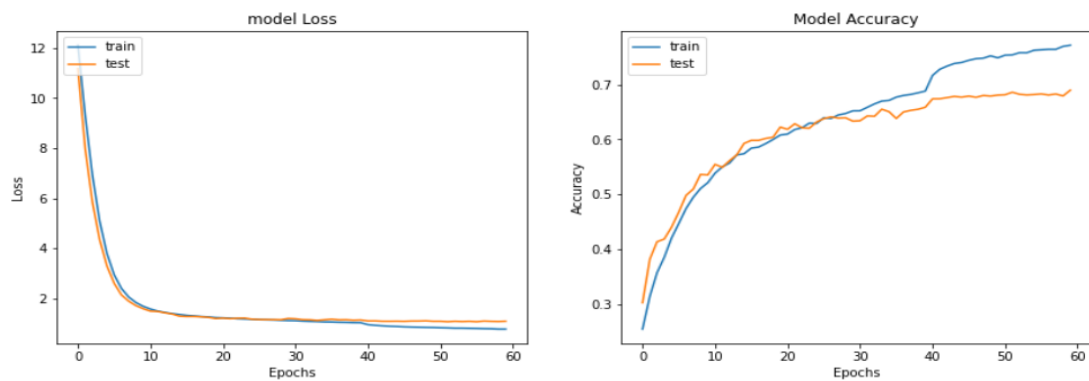


Figure 5. 1Accuracy and Loss

MODEL: SVM AND CNN COMPARISON

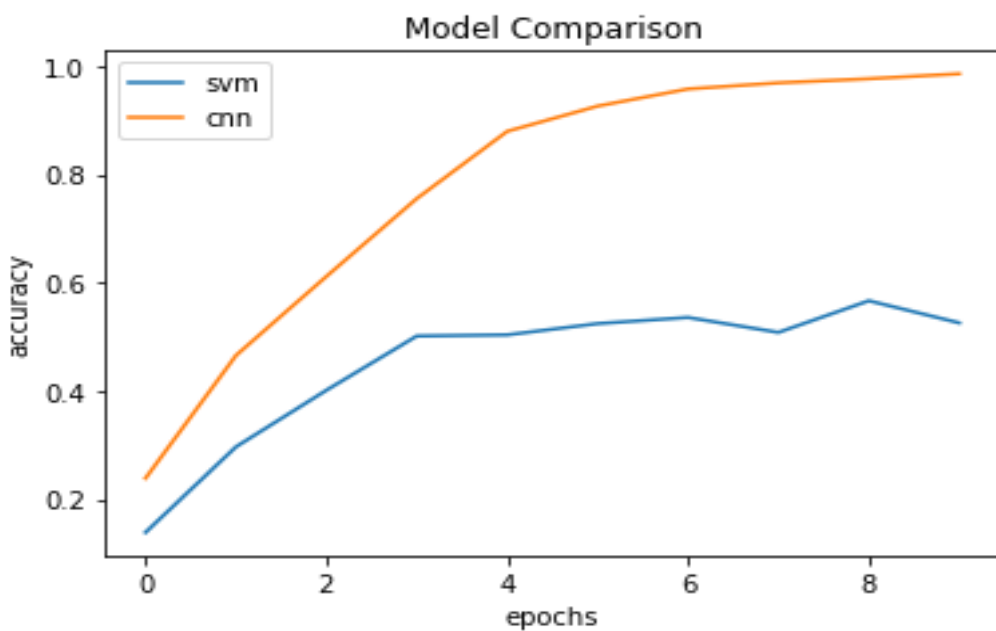
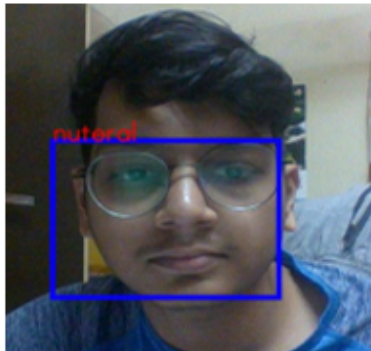


Figure 5. 1SVM vs CNN Comparison

USER IMAGE AND OUTPUTS:

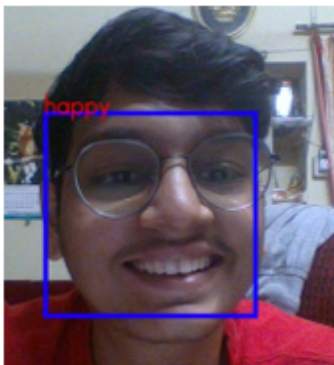
TEST IMAGE1:



Emotion Detected :Neutral
Brazen
The Power of the Dog
The King's Daughter
Licorice Pizza

Figure 5. 3

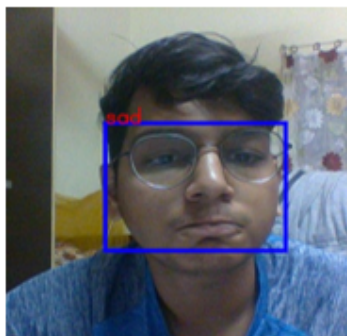
TEST IMAGE2:



Emotion Detected :Happy
Scream
Brazen
The Last Duel
The Tragedy of Macbeth

Figure 5. 4

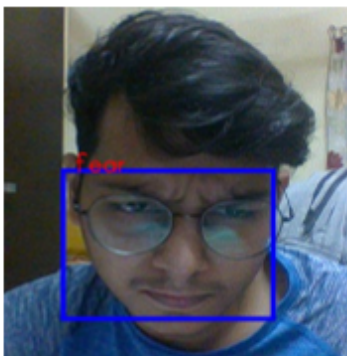
TEST IMAGE 3:



Emotion Detected :Sad
Encanto
Don't Look Up
Hotel Transylvania: Transformania
Sing 2

Figure 5. 5

TEST IMAGE 4:



Emotion Detected :Fear
Encanto
Don't Look Up
Brazen
The Batman
The Last Duel

Figure 5. 6

CHAPTER 6

SUMMARY AND CONCLUSION

Emotion Based Movie Recommender System has the novelty of incorporating user emotions into the user profile to provide users with wellrecommended products based on their emotional state. Because movie is a complex and subjective domain, it is necessary to incorporate user emotion into the user profile. Users can give their feedback about how a recommended movie meets their preferences. This feedback (in the form of user's rating) improves the recommendation quality over time. Last but not least, the system explains to the users the movie recommendations in plain English, describing the choices and the underlying reasoning. The ability to recognize emotion is an aspect of human intelligence that has been argued to be even more important than mathematical and verbal intelligences. Because emotion can influence interactions, behaviours and thinking of the user, we believe that Emotion Based Movie Recommender System with the Emotion detector can greatly improve the efficiency of the movie recommendation. However, the precision of this module should be validated. Moreover, the relation between colours and emotions depends on culture, religion of user. We shall consider these aspects in a future work, as well as the system implementation and validation.

APPENDIX

SOURCE CODE:

```
import tensorflow as tf
from tensorflow import keras
import numpy as np

from google.colab import drive
drive.mount("/content/gdrive/")

import pandas as pd
my_set=pd.read_csv('/content/gdrive/MyDrive/Colab Notebooks/fer2013.csv')

import sys, os
import pandas as pd
import numpy as np

from keras.models import Sequential
from keras.layers import Dense, Dropout, Activation, Flatten
from keras.layers import Conv2D, MaxPooling2D, BatchNormalization, AveragePooling2D
from keras.losses import categorical_crossentropy
from tensorflow.keras.optimizers import Adam
from keras.regularizers import l2
from keras.utils import np_utils

df=my_set

X_train,train_y,X_test,test_y=[],[],[],[]
```

```

for index, row in df.iterrows():
    val=row['pixels'].split(" ")
    try:
        if 'Training' in row['Usage']:
            X_train.append(np.array(val,'float32'))
            train_y.append(row['emotion'])
        elif 'PublicTest' in row['Usage']:
            X_test.append(np.array(val,'float32'))
            test_y.append(row['emotion'])
    except:
        continue

```

```

X_train = np.array(X_train,'float32')
train_y = np.array(train_y,'float32')
X_test = np.array(X_test,'float32')
test_y = np.array(test_y,'float32')

```

```

X_train -= np.mean(X_train, axis=0)
X_train /= np.std(X_train, axis=0)

```

```

X_test -= np.mean(X_test, axis=0)
X_test /= np.std(X_test, axis=0)

```

```

X_train = X_train.reshape(X_train.shape[0], 48, 48, 1)

```

```

X_test = X_test.reshape(X_test.shape[0], 48, 48, 1)

```

```

def build_model1(hp):
    model = keras.Sequential([
        keras.layers.Conv2D(
            filters=hp.Int('conv_1_filter', min_value=32, max_value=64, step=8),

```

```

        kernel_size=hp.Choice('conv_1_kernel', values = [3,5]),
        activation='relu',
        input_shape=(48,48,1)
    ),
    keras.layers.Conv2D(
        filters=hp.Int('conv_2_filter', min_value=32, max_value=64, step=8),
        kernel_size=hp.Choice('conv_2_kernel', values = [3,5]),
        activation='relu'
    ),
    keras.layers.Conv2D(
        filters=hp.Int('conv_3_filter', min_value=64, max_value=128, step=8),
        kernel_size=hp.Choice('conv_3_kernel', values = [3,5]),
        activation='relu'
    ),
    keras.layers.Flatten(),
    keras.layers.Dense(
        units=hp.Int('dense_1_units', min_value=32, max_value=128, step=8),
        activation='relu'
    ),
    keras.layers.Dense(10, activation='softmax')
])

model.compile(optimizer=keras.optimizers.Adam(hp.Choice('learning_rate', value
s=[1e-2, 1e-3])),
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

return model

```

```

def build_model(hp):
    model = keras.Sequential([

```

```

keras.layers.Conv2D(
    filters=hp.Int('conv_1_filter', min_value=64, max_value=128, step=8),
    kernel_size=hp.Choice('conv_1_kernel', values = [3,5]),
    activation='relu',
    input_shape=(48,48,1)
),
keras.layers.Conv2D(
    filters=hp.Int('conv_2_filter', min_value=32, max_value=64, step=8),
    kernel_size=hp.Choice('conv_2_kernel', values = [3,5]),
    activation='relu'
),
keras.layers.Conv2D(
    filters=hp.Int('conv_3_filter', min_value=32, max_value=64, step=8),
    kernel_size=hp.Choice('conv_3_kernel', values = [3,5]),
    activation='relu'
),
keras.layers.Flatten(),
keras.layers.Dense(
    units=hp.Int('dense_1_units', min_value=64, max_value=128, step=8),
    activation='relu'
),
keras.layers.Dense(10, activation='softmax')
])

model.compile(optimizer=keras.optimizers.Adam(hp.Choice('learning_rate', value
s=[1e-2, 1e-3])),
    loss='sparse_categorical_crossentropy',
    metrics=['accuracy'])

return model

```

```
tuner_search=RandomSearch(build_model1,  
                           objective='val_accuracy',overwrite = True,  
                           max_trials=2,directory='output',project_name="try8")
```

```
tuner_search.search(X_train,train_y,epochs=2,validation_split=0.1)
```

```
model=tuner_search.get_best_models(num_models=1)[0]  
model.summary()
```

```
cnn=model.fit(X_train, train_y,  
              batch_size=60,  
              epochs=10,  
              verbose=1,  
              validation_data=(X_test, test_y),  
              shuffle=True)
```

Saving Model

```
fer_json = model.to_json()  
with open("fer.json", "w") as json_file:  
    json_file.write(fer_json)  
model.save_weights("fer.h5")
```

Plotting Results

```
import matplotlib.pyplot as plt
```

```
plt.plot(svm["accuracy"])  
plt.plot(cnn["accuracy"])  
plt.title("Model Comparison")
```

```
plt.ylabel("accuracy")
plt.xlabel("epochs")
plt.legend(["svm", "cnn"], loc="upper left")
plt.show()
```

Detecting Real-Time Emotion

```
import os
import cv2
import sys
import numpy as np
from keras.models import model_from_json
from keras.preprocessing import image
#load model
model = model_from_json(open("fer.json", "r").read())
#load weights
model.load_weights('fer.h5')
face_haar_cascade = cv2.CascadeClassifier('C:/Users/Prakhar Sahu/AppData/Local/Programs/Python/Python39/Lib/site-packages/cv2/data/haarcascade_frontalface_default.xml')
# grab the input image

cam = cv2.VideoCapture(0)

cv2.namedWindow("test")

img_counter = 0

while True:
    ret, frame = cam.read()
    if not ret:
        print("failed to grab frame")
```

```

        break
cv2.imshow("test", frame)

k = cv2.waitKey(1)
if k%256 == 27:
    # ESC pressed
    print("Escape hit, closing...")
    break
elif k%256 == 32:
    # SPACE pressed
    img_name = "opencv_frame_{}.png".format(img_counter)
    cv2.imwrite(img_name, frame)
    print("{} written!".format(img_name))
    img_counter += 1

cam.release()

cv2.destroyAllWindows()
# put the image into the model

test_img = cv2.imread('opencv_frame_0.png',1)
gray_img= cv2.cvtColor(test_img, cv2.COLOR_BGR2GRAY)
faces_detected = face_haar_cascade.detectMultiScale(gray_img, 1.32, 5)

emotions = ('Angry', 'Disgust', 'Fear', 'Happy', 'Sad', 'Surprise', 'Neutral')

for (x,y,w,h) in faces_detected:
    cv2.rectangle(test_img,(x,y),(x+w,y+h),(255,0,0),thickness=7)
    roi_gray=gray_img[y:y+w,x:x+h]#cropping region of interest i.e. face area from image
    roi_gray=cv2.resize(roi_gray,(48,48))

```

```
img_pixels = image.img_to_array(roi_gray)
img_pixels = np.expand_dims(img_pixels, axis = 0)
img_pixels /= 255
```

```
predictions = model.predict(img_pixels)
```

```
#find max indexed array
max_index = np.argmax(predictions[0])
```

```
predicted_emotion = emotions[max_index]
```

Recommender System

```
from bs4 import BeautifulSoup as SOUP
```

```
import re
```

```
import requests as HTTP
```

```
# Main Function for scraping
```

```
def main(emotion):
```

```
# IMDb Url for Drama genre of
```

```
# movie against emotion Fear or Disgust
```

```
if(emotion == "Fear" or emotion == "Disgust"):
```

```
    urlhere = 'http://www.imdb.com/search/title?genres=drama&title_type=feature  
&sort=moviemeter, desc'
```

```
# IMDb Url for Thriller genre of
```

```
# movie against emotion Happy or Surprise
```

```
elif(emotion == "Happy" or emotion == "Surprise"):
```

```
urlhere = 'http://www.imdb.com/search/title?genres=thriller&title_type=feature
&sort=moviemeter, asc'
```

```
# IMDb Url for Comedy genre of
```

```
# movie against emotion Sad or Angry
```

```
elif(emotion == "Sad" or emotion == "Angry"):
```

```
urlhere = 'http://www.imdb.com/search/title?genres=comedy&title_type=featur
e&sort=moviemeter, asc'
```

```
# IMDb Url for Romance genre of
```

```
# movie against emotion Neutral
```

```
elif(emotion == "Neutral"):
```

```
urlhere = 'http://www.imdb.com/search/title?genres=romance&title_type=featu
re&sort=moviemeter, asc'
```

```
# HTTP request to get the data of
```

```
# the whole page
```

```
response = HTTP.get(urlhere)
```

```
data = response.text
```

```
# Parsing the data using
```

```
# BeautifulSoup
```

```
soup = SOUP(data, "lxml")
```

```
# Extract movie titles from the
```

```
# data using regex
```

```
title = soup.find_all("a", attrs = {"href" : re.compile(r'\title\vt+\d*\V')})
```

```
return title
```

```
# Driver Function
```

```

if __name__ == '__main__':

    emotion = predicted_emotion.capitalize()
    a = main(emotion)
    count = 0
    print("Emotion Detected : " +emotion)
    if(emotion == "Disgust" or emotion == "Anger"
        or emotion=="Surprise"):

        for i in a:

            # Splitting each line of the
            # IMDb data to scrape movies
            tmp = str(i).split('>')

            if(len(tmp) == 3):
                print(tmp[1][-3])

            if(count > 13):
                break
            count += 1
    else:
        for i in a:
            tmp = str(i).split('>')

            if(len(tmp) == 3):
                print(tmp[1][-3])

            if(count > 11):
                break
            count+=1

```

OUTPUT:

```
Emotion Detected :Neutral  
Licorice Pizza  
The Power of the Dog  
Marry Me  
The French Dispatch
```

Figure 6. 1 Output Prediction

References

- [1] Burke, R. "Recommender Systems: Survey and Experiments", In User Modeling and User Adapted Interaction. Vol 12. Kluwer Academic Publishers, 2002, pp. 2- 40.
- [2] Burke, R. "Knowledge-based Recommender Systems". In: A. Kent (ed.): Encyclopedia of Library and Information Systems. Vol 69. 2000, pp. 2-23.
- [3] Schafer, J. B. Konstan, J. A. Riedl, J. "ECommerce Recommendation Applications", In Data Mining and Knowledge Discovery. Vol 5. 2001, pp.7-39
- [4] P. Resnick, N. Iakovou, M. Sushak, P. Bergstrom, and J. Riedl, "GroupLens: An Open Architecture for Collaborative Filtering of Netnews," Proc. 1994 Computer Supported Cooperative Work Conf., 1994.
- [5] Matthew Garden, Gregory Dudek. "Mixed Collaborative and Content-Based Filtering with User-Contributed Semantic Features". In, AAAI National Conference on Artificial Intelligence, Boston, Massachusetts. 2006, pp. 1-6
- [6] Rosalind W. Picard, Elias Vyzas, and Jennifer Healey "Toward machine emotional intelligence: analysis of affective physiological state" .In, IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol 23 .2001, pp. 2-4
- [7] Hill, W., Stead, L., Rosenstein, M., And Furnas, G. W "Recommending and evaluating choices in a virtual community of use". In Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems. 1995, pp 194-201.
- [8] Goleman. Emotional intelligence. New York : Bantam Books, 1995. 7
- [9] Ortony and Turner. What's basic about basic emotions? : Psychological review, 1990.
- [10] Oatley, K and Jenkins, J.M. Understanding Emotions: Blackwell, 1996.
- [11] Cacioppo, J. T., Bernston, G. G., Larsen, J. T., Poehlmann, K. M., Ito, T. A. The psychophysiology of emotion. New York: The Guilford Press, 2000.

- [12] Parrott, W. Emotions in Social Psychology. Psychology, Philadelphia, 2001.
- [13] Hemphill, M. A note on adults' color-emotion associations . Journal of Genetic Psychology, 1996.
- [14] Mahnke F. Color, environment, human response. New York: Van Nostrand Reinhold, 1996.
- [15] Black, J. "Color Psychology for E-book Cover Design". 2002.
- [16] Wexner. L.B. The degree to which colors are associated with mood-tones. J. appl. Psychol.1954.
- [17] Murray , D. C., Deabler, H.L. Colors and moodtones. Journal of applied Psychology. 1957.
- [18] Schaie, K. W. Scaling the association between colors and mood tones. The American Journal of Psychology.1961.
- [19] Wexner, L. B. The degree to which colors are associated with mood-tones. Journal of Applied Psychology. 1982.
- [20] Collier, G. L. Affective synaesthesia: Extracting emotion space from simple perceptual stimuli. Motivation and Emotion. 1996.
- [21] Ballast, D. K. Interior design reference manual. Professional Pub. Inc.: Belmont, CA, 2002.
- [21] Kaya, N. Epps, H. Relationship between color and emotion: A study of college students. College Student Journal. 2004.
- [22] Odber, H. S., Karwoski, T. F., Eckerson, A. B. "Studies in synesthetic thinking: I. Musical and verbal associations of color and mood." In The Journal of General Psychology. 1942. pp 26, 153- 173.
- [23] Frasson, C., Chaffar, S., Abdel Razek M., Ochs M. "Using Machine-Learning Techniques to Recognize Emotions for On-Line Learning." In The 18th International FLAIRS Conference, AAAI Press, Clearwater, FL, USA. 2005.pp 15-17.

- [24] Pecjak, V. "Verbal synesthesiae of colors, emotions, and days of the week." In *Journal of Verbal Learning and Verbal Behavior*. 1970. pp 9, 623-626.
- [25] Boyatzis, C. J., Varghese, R. "Children's emotional associations with colors." *Journal of Genetic Psychology*. 1994. pp 155, 77-85.
- [26] Cover T. M. and Hart, P. E. "Nearest neighbor pattern classification." *IEEE Transaction on Information Theory*. 1967. IT-13:21-27.
- [27] Resnick, P. Iacovou, N., Sushak, N., Bergstrom, P. and Riedl, J.. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 Computer Supported Collaborative Work Conference*, pages 175-186, Chapel Hill, North Carolina, 1994.
- [28] Herlocker, J. L. Konstan, J. A. Riedl, J. Explaining collaborative filtering recommendations, *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, 2000.
- [29] Peppers, D. Rogers, M. *The One to One Future: Building Relationships One Customer at a Time*, Currency Doubleday, 1997.
- [30] McNee, S.M. Riedl, J. and Konstan, J.A. Being Accurate is Not Enough: How Accuracy Metrics have hurt Recommender Systems. In the *Extended Abstracts of the 2006 ACM Conference on Human Factors in Computing Systems (CHI 2006)*, Montreal, Canada, April 2006