Kathmandu University

Department of Computer Science and Engineering

COMP 472: Artificial Intelligence

Facial expression Detection

Submitted by:

Julesh Shrestha (44)

Pratap Shrestha (47)

Submitted to:

Santosh Khanal

Assistant Professor

DoCSE

Submission Date: 11th March, 2020

Introduction:

Facial Expression Detection based on artificial intelligence is a software application that finds human beings' expressions. This is the project that detects emotions of people from the webcam/camera. This is expected to create models up to 7 different emotions from human beings. Though studies are far away from excellent results even today. The topic is still satisfying.

On a day to day basics humans commonly recognize emotions by characteristic features, displayed as a part of a facial expression. For instance, happiness is undeniably associated with a smile or an upward movement of the corners of the lips. Similarly, other emotions are characterized by other deformations typical to a particular expression. Research into automatic recognition of facial expressions addresses the problems surrounding the representation and categorization of static or dynamic characteristics of these deformations of face pigmentation

Objectives:

1. To detect emotions made by human beings through application of AI.
2. To train data sets that helps to predict expression.
3. To implement Convolutional Neural Networks for classification of facial expressions.

Agent Description:

PEAS stands for Performance, Environment, Actuators, and Sensors. Based on these properties of an agent, they can be grouped together or can be differentiated from each other. Each agent has the following properties defined for it.

- **Performance**
  The output which we get from the agent. All the necessary results that an agent gives after processing comes under its performance.

- **Environment:**
  All the surrounding things and conditions of an agent fall in this section. It basically consists of all the things under which the agents work.

- **Actuators**:
  The devices, hardware or software through which the agent performs any actions or processes any information to produce a result are the actuators of the agent.

- **Sensors:**
  The devices through which the agent observes and perceives its environment are the sensors of the agent.

Performance	**Accuracy of detecting facial expression correctly or not**
Environment	**Video frame with human face**
Actuators	**Display screen**
Sensors	**Camera, Keyboard**

Agent environment:

The environment is where agents live, operate and provide the agent with something to sense and act upon it. According to Norvig, the environment can have various features from the point of view of an agent.

1. Fully observable vs Partially Observable:

- If an agent sensor can sense or access the complete state of an environment at each point of time then it is a fully observable environment, else it is partially observable.
- Partially observable environment is one in which the agent can never see the entire state of the environment.

  Facial Expression Detection Agent is **Partially Observable** as the result is only shown in frame and the entire state is unknown to the face detection agent.

2. Deterministic vs Stochastic:

- If an agent's current state and selected action can completely determine the next state of the environment, then such an environment is called a deterministic environment.
- A stochastic environment is random in nature and cannot be determined completely by an agent.

  Facial Expression Detection Agent is **Stochastic** as this project is based on real world environment and real-world environments are stochastic in nature.

3. Episodic vs Sequential:

- In an episodic environment, there is a series of one-shot actions, and only the current percept is required for the action.
- However, in a Sequential environment, an agent requires the memory of past actions to determine the next best actions.

  Facial Expression Detection Agent is **Sequential** as it requires trained data sets to determine facial expression.

4. Single-agent vs Multi-agent

- If only one agent is involved in an environment, and operating by itself then such an environment is called a single agent environment.
- However, if multiple agents are operating in an environment, then such an environment is called a multi-agent environment.

  Facial Expression Detection Agent is a **Single Agent.**

5. Static vs Dynamic:

- If the environment can change itself while an agent is deliberating then such an environment is called a dynamic environment.
- A static environment is unchanged when an agent is reflecting on it.

  Facial Expression Detection Agent is **Dynamic** because the result shown will be depending upon the face that is given as input by the user as environment.

6. Discrete vs Continuous:

- If in an environment there are a finite number of percepts and actions that can be performed within it, then such an environment is called a discrete environment else it is called a continuous environment.

  Facial Expression Detection Agent is **Continuous** as it will be showing result continuously and calculating results at the real time.

Problem specification:

Human emotions and intentions are expressed through facial expressions and deriving an efficient and effective feature is the fundamental component of facial expression system. Facial expressions convey non-verbal cues, which play an important role in interpersonal relations. Automatic recognition of facial expressions can be an important component of natural human machine interfaces; it may also be used in behavioral science and in clinical practice. An automatic Facial Expression Recognition system needs to solve the following problems: detection and location of faces in a cluttered scene, facial feature extraction, and facial expression classification.

Data source:

The dataset from a Kaggle Facial Expression Recognition Challenge (FER2013) is used for the training and testing. It comprises pre-cropped, 48-by-48-pixel grayscale images of faces each labeled with one of the 7 emotion classes: anger, disgust, fear, happiness, sadness, surprise, and neutral. Dataset has training set of 35887 facial images with facial expression labels.. The dataset has class imbalance issue, since some classes have large number of examples while some has few. The dataset is balanced using oversampling, by increasing numbers in minority classes. The balanced dataset contains 40263 images, from which 29263 images are used for training, 6000 images are used for testing, and 5000 images are used for validation.

Algorithm used:

Fisherface (Face recognition algorithm)

The Fisherface algorithm learns a class-specific transformation matrix, so they do not capture illumination as obviously as the Eigenfaces method. It is especially useful when facial images have large variations in illumination and facial expression. The Discriminant Analysis instead finds the facial features to discriminate between the persons. It's important to mention, that the performance of the Fisherfaces heavily depends on the input data as well.

The input of a face recognition is image or video stream and the output is an identification or verification of the subject or subjects that appear in the image or video.

CNN

The facial expression recognition system is implemented using convolutional neural networks. The block diagram of the system is shown in following figures:

Fig: Training Phase

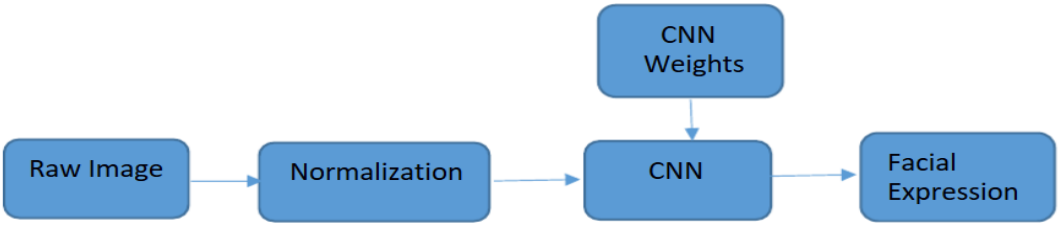

Fig: Testing Phase

During training, the system receives training data comprising grayscale images of faces with their respective expression label and learns a set of weights for the network. The training step took as input an image with a face. Thereafter, an intensity normalization is applied to the image. The normalized images are used to train the Convolutional Network. To ensure that the training performance is not affected by the order of presentation of the examples, validation dataset is used to choose the final best set of weights out of a set of trainings performed with samples presented in different orders. The output of the training step is a set of weights that achieve the best result with the training data. During test, the system received a grayscale image of a face from test dataset, and output the predicted expression by using the final network weights learned during training. Its output is a single number that represents one of the seven basic expressions

Libraries Used:

- **NumPy:** NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

- **openCV:** OpenCV is a library of programming functions mainly aimed at real-time computer used for Image Processing. It is mainly used to do all the operations related to Images.

- **Keras:** Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation.

Outcome analysis:

Emotions stored as numerical as labeled from 0 to 6. Keras would produce an output array including these 7 different emotion scores. We can visualize each prediction as bar chart.

The following picture of Pablo Escobar is taken in a police station when he was taken into custody. It seems that the model we've constructed can successfully recognize Pablo in happy mood.

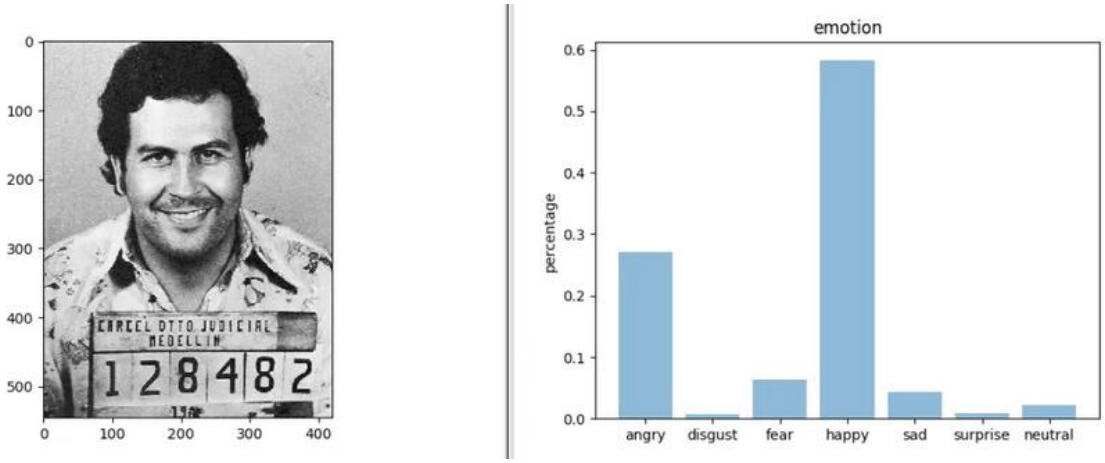

Fig: Pablo Escobar's facial expression

The network says that Mona Lisa is in neutral mood.

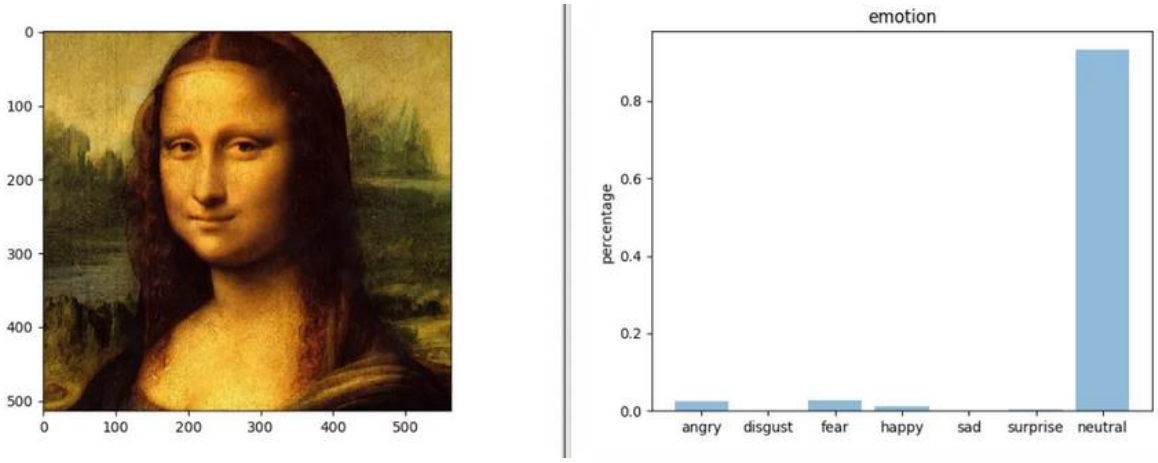

Fig: Da Vinci's Mona Lisa's facial expression'

Accuracy:

Training image batch size was taken as 30, while filter map is of size 20x5x5 for both convolution layer. Validation set was used to validate the training process. In last batch of every epoch in validation cost, validation error, training cost, training error are calculated. Input parameters for training are image set and corresponding output labels. The training process updated the weights of feature maps and hidden layers based on hyper-parameters such as learning rate, momentum, regularization and decay. In this system batch-wise learning rate was used as 10e-5, momentum as 0.99, regularization as 10e-7 and decay as 0.99999. The testing of the model is carried out using 6000 images. The classifier provided 56.77 % accuracy. This model has highest accuracy for disgust emotion with 95.23 %, followed by happy with 68.86 %, surprise with 64.52 %, neutral with 49.31 %, anger with 42.16 %, fear with 38.31 % and lowest accuracy for sad emotion as 38.23 %.

The model performs really well on classifying positive emotions resulting in relatively high precision scores for happy and surprised. Disgust has highest precision and recall as 0.95 and 0.99 as images in this class were oversampled to address class imbalance. Happy has a precision of 0.68 and recall of 0.69 which could be explained by having the most examples (6500) in the training set. Interestingly, 16 surprise has a precision of 0.69 and recall of 0.65 having the least examples in the training set. There must be very strong signals in the surprise expressions. Model performance seems weaker across negative emotions on average. In particularly, the emotion sad has a low precision of only 0.44 and recall 0.38. The model frequently misclassified angry, fear and neutral as sad. In addition, it is most confused when predicting sad and neutral faces because these two emotions are probably the least expressive (excluding crying faces).

CNN Classifier is then used to classify image taken from webcam in Laptop. Face is detected in webcam frames using Haar cascade classifier from OpenCV. Then detected face is cropped and normalized and fed to CNN Classifier.

Conclusion

In this project, six-layer convolution neural network is implemented to classify human facial expressions i.e. happy, sad, surprise, fear, anger, disgust, and neutral. The system has been evaluated using Accuracy, Precision, Recall and F1-score. The classifier achieved accuracy of 56.77 %, precision of 0.57, recall 0.57 and F1-score 0.57.

References

1. Face Recognition with OpenCV¶. (n.d.). Retrieved from https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html

2. Keras: The Python Deep Learning library. (n.d.). Retrieved from https://keras.io/

3. Martinez, A. (n.d.). Fisherfaces. Retrieved from http://www.scholarpedia.org/article/Fisherfaces

4. Serengil, S. (2018, March 12). A Gentle Introduction to Convolutional Neural Networks. Retrieved from https://sefiks.com/2017/11/03/a-gentle-introduction-to-convolutional-neural-networks/