Machine Learning Engineer Nanodegree

Capstone Proposal

Shvet Chakra December 3rd, 2020

Proposal

Emotion recognition using facial expressions

Domain Background

Facial expressions and related changes in facial patterns give us information about the emotional state of the person and help to regulate conversations with the person. Moreover, these expressions help in understanding the overall mood of the person in a better way. Facial expressions play an important role in human interactions and non-verbal communication. Classification of facial expressions could be used as an effective tool in behavioural studies and in medical rehabilitation. Facial expression analysis deals with visually recognizing and analyzing different facial motions and facial feature changes.[1]

Motivation Nowadays, when technology has penetrated into the personal life of humans with mobile devices and interaction of machines with humans is carried out on the daily basis, it is getting important for the machine to recognize the emotion of the fellow human while interating with him. Recognition of emotion can help machines powered with artificial intelligence to enhance this interaction. Possible application where this project can be helpful in wellbeing of humans. As recognition & acceptance is the first step of any problem, this solution can help the user about their mood and help them in taking appropriate step. For eg. we have facial recognition phone security feature to lock/unlock the device, if the device captures the face and determine the emotion of the users on several intervals and then give a happy mood percentage for the day. It can help them in recognizing about their mood if it is not appropriate and motivate them to take necessary action.

Problem Statement

Facial expressions play an important role in recognition of emotions and are used in the process of non-verbal communication, as well as to identify people. They are very important in daily emotional communication, just next to the tone of voice [2]. They are also an indicator of feelings, allowing a man to express an emotional state [3]. People, can immediately recognize an emotional state of a person. As a consequence, information on the facial expressions are often used in automatic systems of emotion recognition [4]. The aim of the project is to recognize seven basic emotional states: neutral, joy, surprise, anger, sadness, fear and disgust based on facial expressions.[5] Numerous investigators have used neural networks for facial expression classification. The performance of a neural network depends on several factors including the initial random weights, the training data, the activation function used, and the structure of the network including the number of hidden layer neurons, etc. Here I will try to create a CNN based nueral network model to classify the images into their seven basic emotional states: neutral, joy, surprise, anger, sadness, fear and disgust.

Datasets and Inputs

The project has been chosen from a kaggle competion and hence the same datset will be used here also to train and test the model for the proposed problem. https://www.kaggle.com/c/emotion-detection-from-facial- expressions/data

We need to check the images in our dataset for their sizes and number of color components

```
""" Check image size and color components"""
df = pd.DataFrame()

targetdir = "images"
filelist = glob.glob(targetdir+str("/*"))
for file in filelist:  img = cv.imread(file)

    img_shape = img.shape
    #print(img_shape)
    df = df.append(pd.Series(img_shape),ignore_index=True)
df = df.rename(columns={0: "Width", 1: "Height", 2:"Components"}) df.head()
```

Out[3]:

	Width	Height	Components
0	350.0	350.0	3.0
1	350.0	350.0	3.0
2	350.0	350.0	3.0
3	350.0	350.0	3.0
4	350.0	350.0	3.0

Now we need to determine the minimum & maximum width and height from all the image sizes

Df.info()

```
<class      'pandas.core.frame.DataFrame'>
RangeIndex: 13718 entries, 0 to 13717 Data
columns (total 3 columns):

---   ------        --------------   -----
 # Column Non-Null Count Dtype
 1.    Width         13718 non-null      float64
 2.    Height        13718 non-null      float64
 3.    Components    13718 non-null      float64
dtypes: float64(3)

memory usage: 321.6 KB
```

```
df.agg(['max', 'min', 'mean', 'std'])
```

Out[5]:

	Width	Height	Components
max	536.000000	441.000000	3.0
min	24.000000	18.000000	3.0
mean	335.768698	333.548039	3.0
std	58.700772	63.898948	0.0

```
# Load in image data
image_info = pd.read_csv('data/legend.csv',delimiter=',')
image_info.head()

image_info.info()
```

```
<class      pandas.core.frame.DataFrame'>
RangeIndex:  13690  entries,  0  to  13689
```

```
Data columns (total 3 columns):
---  ------  --------------  -----
 # Column Non-Null Count Dtype
1)  user.id   13690 non-null   object
2)  image     13690 non-null   object
3)  emotion   13690 non-null   object
dtypes: object(3)
memory usage: 321.0+ KB
```

Image_info.head()

	user.id	image	emotion
0	628	facial-expressions_2868588k.jpg	anger
1	628	facial-expressions_2868585k.jpg	surprise
2	628	facial-expressions_2868584k.jpg	disgust
3	628	facial-expressions_2868582k.jpg	fear
4	dwdii	Aaron_Eckhart_0001.jpg	neutral

```
image_info["emotion"].unique()
```

```
array(['anger', 'surprise', 'disgust', 'fear', 'neutral','happiness',
       'sadness', 'contempt', 'NEUTRAL', 'SADNESS', 'DISGUST', 'FEAR',
       'SURPRISE', 'ANGER', 'HAPPINESS'], dtype=object)
```

```
image_info["emotion"].value_counts()
```

```
neutral      6717
happiness    5309
HAPPINESS     387
surprise      356
anger         228
DISGUST       195
NEUTRAL       151
SADNESS       144
sadness       124
ANGER          24
fear           13
disgust        13
SURPRISE       12
contempt        9
FEAR            8
Name: emotion, dtype: int64
```

We can see the various emotion categories and there distribution in the dataset.

```python
# converting the emotion string to lowercase
image_info["emotion"] = image_info["emotion"].str.lower()
image_info["emotion"].value_counts()
```

```
neutral      6868
happiness    5696
surprise      368
sadness       268
anger         252
disgust       208
fear           21
contempt        9
Name: emotion, dtype: int64
```

```python
""" Merge the contempt into disgust category"""
image_info["emotion"] = image_info["emotion"].replace("contempt","disgust")
```

```python
image_info["emotion"].value_counts()
```

```
neutral      6868
happiness    5696
surprise      368
sadness       268
anger         252
disgust       217
fear           21
Name: emotion, dtype: int64
```

It can be easily noticable that the emotion categories are not equally distributed here. Hence we need to drop some images from neutral and happiness category to make the distribution even for all categories.

```python
#move above selected images into a separate dataset folder
neutral = 0
happy = 0
selectedImages = pd.DataFrame(columns=["image","emotion"])
for index, row in image_info.iterrows():
    #if happy >= 500 and neutral >=500:
     #    break
    if row["emotion"] == "happiness": happy += 1

        if happy > 500:
            continue
    elif row["emotion"] == "neutral": neutral += 1

        if neutral > 500:
            continue
    selectedImages = selectedImages.append(pd.Series(row),ignore_index=True)
    #print(row)
```

```
selectedImages["emotion"].value_counts()
```

```
neutral      500
happiness    500
surprise     368
sadness      268
anger        252
disgust      217
fear          21
Name: emotion, dtype: int64
```

Now we have almost balanced the distribution of different category images and use this set to further break it into train set, validation set and test set to perform the training of our model and test set can be used for benchmarking purposes.

Solution Statement

Convolution networks or convolution neural networks are a specialized kind of neural networks for processing data that has a known grid-like topology. Examples inclued time-series data and image data which has a grid like structure. Convolution leverages three important ideas that can help improve a machine learning system; sparse interctions, parameter sharing and equivariant representations. Moreover, convolution provides a means for working with inputs of variable sizes. One major advantage of using CNNs over NNs is that you do not need to flatten the input images to 1D as they are capable of working with image data in 2D. This helps in retaining the "spatial" properties of images.

Benchmark Model

For benchmarking we can use Alexnet or VGG models to perform some benchmarking. https://www.imperial.ac.uk/intelligent-digital-systems/cnn-benchmark-suite/ Our model can also we compared for categorization accuracy as per kaggle competition leaderboard https://www.kaggle.com/c/emotion- detection-from-facial-expressions/leaderboard

Evaluation Metrics

We can use following metrics to evaluate our model

4. Classification Accuracy
5. Confusion Matrix
6. F1 Score

Project Design

Technically, deep learning CNN models to train and test, each input image will pass it through a series of convolution layers with filters (Kernals), Pooling, fully connected layers (FC) and apply Softmax function to classify an object with probabilistic values between 0 and 1.

The project can be mainly clasified into following steps:

4) Load and preprocess the image dataset:

The dataset will be analyzed for the image attributes and the list of emotions labeled to them. Conversion of labels to lowercase and merge few labels into one may also be explored if required. Then we will divide the dataset into three sets, train set, validation set , and test set.

For the training the model, we will apply some general image transformations such as random scaling, cropping, and flipping.
The input data is center cropped to or resized to 224x224 pixels as required by the pre-trained networks for benchmarking.
Next no scaling or rotation transformations will be performed on validation set and test set but need to resize then crop the images to the appropriate size to measure the model's performance on data it hasn't seen yet.

5) Building and training the classifier

Now that the data is ready, it's time to build and train the classifier. As usual, we should use one of the pretrained models from torchvision.models to get the image features. Build and train a new feed-forward classifier using those features.

5.1) Load a pre-trained network

5.2) Define a new, untrained feed-forward network as a classifier, using ReLU activations and

dropout

5.3) Train the classifier layers using backpropagation using the pre-trained network to get the

features

5.4) Track the loss and accuracy on the validation set to determine the best hyperparameters

6) Evaluate the model on the test set and evaluation metrics

We can use following metrics to evaluate our model

1. Classification Accuracy
2. Confusion Matrix
3. F1 Score

References:

[1] https://biomedical-engineering-online.biomedcentral.com/articles/10.1186/1475-925X-8-16 [2] Ratliff M. S., Patterson E., Emotion recognition using facial expressions with active appearance models, Proceedings of the Third IASTED International Conference on Human Computer Interaction, ACTA Press, Anaheim, CA, USA, 2008, 138–143. [3] Tian Y. I., Kanade T., Cohn J. F., Recognizing action units for facial expression analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence, 23 (2001), no. 2, 97–115. [4] Mao Q., Pan X., Zhan Y., Shen X., Using Kinect for real-time emotion recognition via facial expressions, Frontiers Inf Technol Electronic Eng, 16 (2015), no. 4, 272–282. [5]International Conference on Computational Science, ICCS 2017, 12-14 June 2017,Zurich, Switzerland Emotion recognition using facial expressions, Paweł Tarnowski, Marcin Kołodziej, Andrzej Majkowski, Remigiusz J. Rak