Emotion Detection

Şeyma Yılmaz
Hacettepe University
Beytepe, ANKARA
seymayilmaz@hacettepe.edu.tr

Mücahit Fındık
Hacettepe University
Beytepe, ANKARA
mucahitfindik@hacettepe.edu.tr

Yusuf Emre Genç
Hacettepe University
Beytepe, ANKARA
yusufemregenc@hacettepe.edu.tr

Abstract

In this paper, CNN based project which classifying people's emotions using their images will be mentioned. In the project, we try to understand people's emotions by using facial expressions. The reason we use people's facial expressions, we know that they use facial expressions rather than verbal to express their feelings and facial expressions are mostly used way of expressing among non-verbal expression ways. To create the model, we use data sets consist of humans and animations. The size of images is 48x48. When developing our model we use the CNN algorithm, which is very suitable for classifying images. We use the confusion matrix to measure the performance of our machine learning classification problem. By this means, we measure how successful the model we are using according to the data set we use, and we make our model more appropriate according to the information.

Besides, we believe that the study will be the basis for many projects in the future. Our project may be used in new smart cars to increase the driver's focus by music recommendation or in a session to increase the productivity of the speech by giving a break. Finally, we test our model in real-time using the model we created. Thus, we prove the usability of the system.

1. Introduction

People express their thoughts and emotions not only verbal but also non-verbal. Albert Mehrabian who is a psychology professor found in his observations that non-verbal behaviors are more effective rather than what people say [1]. Trying to understand non-verbal behaviors is important in helping to understand what is actually happening during the conversation [2]. Facial expressions come first among non-verbal behaviors.

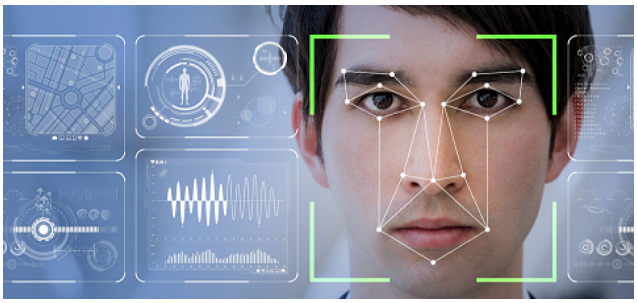

Figure 1. Emotion Detection.
(taken from https://www.marketreportgazette.com/emotion-detection-and-recognition-market-2019-26-stringent-regulations-and-increasing-adoption-by-affectiva-emotient-eyeris-kairosnoldus).

Facial expressions are very important for clearly expressing how we feel to people and to clearly understand what they are feeling. From the moment people were born, they began to express their troubles with facial expressions. Then, they continued to learn to express their emotions and understand the emotions of people by using facial expressions throughout their lives. In doing all that, people use their intuitions and observations. In light of all this information, we wanted to detect emotions from facial expressions with the help of machine learning algorithms. In this project, we aim the machines to perceive the emotions that people perceive with their intuition and observations.

One of the main contributions of this project is to provide infrastructure for many important projects by classifying people's emotions according to their pictures. For example, when conducting psychological analyzes, facial expressions of people are of great importance. Using a machine-

learning algorithm to determine the emotion of a person in a picture makes these investigations independent of the intuition of the researcher. Thus, more objective results are achieved in a faster research process. Another example is that an assistant who detects moments when people are unhappy. It can plays music for them or opens entertaining videos and so it keeps people away from depressed feelings. For another example, knowing the long-distance bus drivers' emotions may prevent disasters by recommending them positive playlists. Emotion classification can be used in many imaginable projects like these. Another contribution of the project is to transform the subjective comments perceived by the intuition of the people into objective interpretations with the machine learning algorithm, as in the example mentioned above. Besides, even in an environment where people are not present, the emotions of people can be perceived, that is, the emotion perception mechanism can be free from the people.

We used 3 data sets when training our model. Two of them are human data sets and the other is an animation data set. Our data set has 7 basic emotions; angry, disgust, fear, happy, neutral, sad, surprised. We divided our data set into training, test and validation part. All images in the data set designated to grey scale and the size of images are 48x48. We also applied to our data set data augmentation method.

The most challenging part of our project was dealing with the data set. Our first *data set* [3] was very unbalanced and poor. For example, there were a very low number of disgust instances in the data set. Therefore, the results with the first data set were not satisfying. So, we need to research a new data set.

We found a new *data set* which also formed by human images [4]. Our new data set is called as fer2013. We combined two data sets and we had approximately 53k images in the training set, 3k images in the test set and 3k images in the validation set. After combining the data sets, we observed that our result is much better.

We discovered that animation data can be useful to increase accuracy. So we used a *data set* of animations [5]. This data set needed so much time to train. Therefore we add only 4.2k animation images to the training set. This gave us a serious improvement. So the number of images in our data set is approximately 63k.

We measured the accuracy of the models we created between all these transitions using a confusion matrix. Thanks to the confusion matrix, we were able to observe the accuracy of our model very clearly. Besides, we understood the difficulties in the data set. We easily identified which classes are problematic and take various measures such as using data augmentation.

Finally, we tested our model in a real-time environment. For this, we identified the faces in the video and guessed the emotion of the people by using our model. Thus, we proved the usability of our project. In other parts of the page, we will describe in more detail below topics:

In section 2, other important works similar to our project topic,
In section 3, the technical details about our project work,
In section 4, some experiments in which we analyze the performance of the approaches,
In section 5, a summary of all our project work.

2. Related Works

Examining people's facial expressions is very popular in machine learning. So, many studies have also been conducted to determine emotion in people's facial expressions. In this section, we will talk about some of the studies we have examined.

The first paper is *Face detection and recognition of natural human emotion using Markov random fields*. In the project, three modules are used. The first module uses Markov random fields (abbreviated as MRF). By using MRF, the model detects the skin and segments images. The second module finds the location of the eyes and mouth. In the last module, emotion is detected by using edge detection and the location data of eyes and mouth.

The second paper is *Facial emotion recognition using multi-modal information*. There is another important way, as well as image, to detect emotion that is auditory data. This idea makes the project multi-model. This hybrid model is trained by video recordings. In the data set collection phase, 36 different emotional sentences of two speakers were recorded. Afterward, a multi-model was created using these video recordings.

The third paper is *Emotional state and the detection of change in facial expression of emotion*. In this article, the usage areas of the model that we want to develop are given. One of them showed during the experiment that the emotional scenes discovered by the films had a greater impact on the audience. Another area of usage is that the speaker changes the flow of the speech by taking into account the emotion that the listener is getting from facial expressions.

The fourth paper is *Emotion Detection Through Facial Feature Recognition*. In the project, emotion is detected through facial feature recognition. Viola-Jones and Harris algorithms are used. Viola-Jones helps to detect faces in a way that's so accurate and fast [5]. Harris algorithm is used in lots of computer vision problems. It helps to extract the edges from the image. This algorithm is used to detect the edge of the face which caters to calculate slopes between eyes and mouth. This project uses principal component analysis, linear discriminant analysis, histogram-of-oriented-gradients (HOG) feature extraction and support vector machines (SVM) to train a multi-class predictor for classifying the seven classes that are angry, disgust, fear,

happy, neutral, sad, surprised.

The fifth paper is *Image based Static Facial Expression Recognition with Multiple Deep Network Learning*. In the paper, they detect emotion with an image-based static facial expression recognition method. The data set they used is Static Facial Expressions in the Wild (SFEW) 2.0 data set. They use a face detection module that contains three state-of-art face detectors. The following module is a CNN module that is pre-trained with data set FER-2013. The models that are pre-trained are fine-tuned on SFEW 2.0. Their model achieves 55.96% and 61.29% respectively on the validation and test set of SFEW 2.0.

3. The Approach

In the project, we used the CNN algorithm, one of the most popular deep learning algorithms. The fact that CNN is a rescuer model for all image-related problems was the most important reason for us to decide on the algorithm. It also simplifies model complexity by using convolution and pooling operations. This makes CNN more efficient in terms of calculation. Another reason is that CNN can detect the most critical features without human oversight, sometimes even better than a human being. For example, in our project, we use CNN to estimate people's emotions based on the posture of people's mouth, nose, and eyes.

We perform convolution and pooling operations for the images in the data set we are working on. Then, we apply the fully connected layer. You can see the layers used in the model in *Figure 2* in more detail.

3.1. Convolution Layer

The convolution layer simplifies the processing of images without losing important features for better results when classifying images and making the model more effective by reducing the number of operations. The size of the images in our data set is 48x48. If we want to use every pixel in these images, we need to use 2304 (48x48) neurons. In addition, the 2304 weight needs to be updated during the back-propagation phase. Considering that similar processes will be repeated in every layer, it will not be difficult to predict that our model will run slowly. In addition, these processes force computer hardware and can cause a long wait for training. To resolve the issue, CNN uses the convolution filter, also called the kernel. Using this kernel matrix, which is smaller than the image matrix, we manipulate the values by applying a mathematical operator to each pixel. Thus, we convert raw pixel values into more meaningful and useful information. The kernel sizes used for each layer:

Conv2D_1: kernel_size = (3, 3)
Conv2D_2: kernel_size = (3, 3)
Conv2D_3: kernel_size = (3, 3)

Layer (type)	Output shape	# parameters
Conv2D_1	46, 46, 32	320
Conv2D_2	44, 44, 64	18496
MaxPooling2D_1	22, 22, 64	0
Conv2D_3	20, 20, 128	73856
MaxPooling2D_2	10, 10, 128	128
Conv2D_4	8, 8, 128	147584
MaxPooling2D_3	4, 4, 128	0
Conv2D_5	4, 4, 7	903
Conv2D_6	1, 1, 7	791
Flatten	7	0
Activation	7	0

Figure 2. Table 1

Conv2D_4: kernel_size = (3, 3)
Conv2D_5: kernel_size = (1, 1)
Conv2D_6: kernel_size = (4, 4)

The convolution operation is carried out by sliding the above-mentioned filters over the input image. After multiplying the matrix between the elements for the position of the filter, the results are added and a pixel of the new feature matrix is created. The operation is repeated for each sliding. For example, there is 48x48 image as input for the first convolution layer and the kernel size is 3x3. When sliding is completed, the output to be used as input in the other layer will be 46x46. As a result, the process reduces the size of the image and makes the image more useful for subsequent layers. This gives us functionality.

3.2. Pooling Layer

After the convolution layer, usually, pooling is applied. The main reason for applying this process is to reduce the matrix size as much as possible. Thus, training time can be reduced. Besides, over-fitting is also avoided. We used the max-pooling method in our project. You can see the working logic of the method in *Figure 3*. Unlike the convolution layer, the pooling layer does not have parameters. A window is slid according to the pool size specified on the input image and the highest value within the window is taken. It is ensured that the windows do not overlap according to the specified number of strides. This significantly reduces the matrix size.

Besides, you can see the dimensions of the matrices before and after the pooling layer according to the pool size specified in *Figure 4*.

3.3. Fully Connected Layer

In the fully connected layer, the output of convolution and pooling are obtained and the image is classified. To do this, the final output from the convolution and pooling layers is flattened as a vector. This vector shows the probabil-

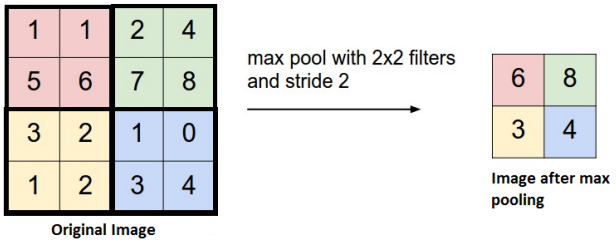

Figure 3. Max Pooling.
(taken from https://www.analyticsvidhya.com/blog/2017/05/25-must-know-terms-concepts-for-beginners-in-deep-learning/pooling/).

Input	Pool size	# stride	Output
44, 44, 64	2,2	2	22, 22, 64
20, 20, 128	2,2	2	10, 10, 128
8, 8, 128	2,2	2	4,4,128

Figure 4. Table 2

ity that the properties belong to classes. Then we decided to use ReLU as the activation function because it requires less mathematical operations than most. Also, one of the important features of ReLU is that it activates multiple neurons at the same time and sparse the network [6]. In *Figure 5*, you can observe the value ranges of the ReLU function.

Furthermore, since it is effective in multiple classifications, the probability of each class being correct is obtained by soft-max applied just before the output layer. In fact, soft-max represents the probability distributions of each class relative to the other classes. That is, all of the calculated probabilities are between 0 and 1, and the sum of these probabilities is equal to 1. You can refer to *Figure 6* for the logic of the softmax function.

4. Experimental Results

In the project, we created our own model for the first time. In addition, we first used a data set containing 37k images of people. As a result, the accuracy of the model was measured as 49%. We thought the reason for the accuracy that did not satisfy us was the low complexity of the model.

Therefore, we thought that the FastAI library would be useful for ready-made models. Because we could set different learning rates in different areas of our model. In addition, data augmentation was more functional and easier in the library. In the library, we used ResNet-34 and DenseNet-201 models. After moving to the FastAI library, we tried to increase our data by using get_transform() method. Using this function, the desired data augmentation method can be used. We used this by default.

When we tried to train ResNet-34 and DenseNet-201

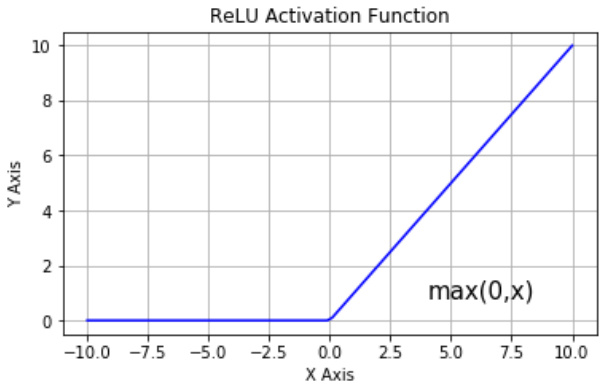

Figure 5. ReLU function.
(taken from https://analyticsindiamag.com/most-common-activation-functions-in-neural-networks-and-rationale-behind-it/).

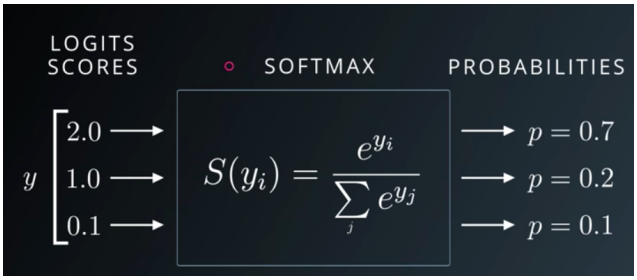

Figure 6. Softmax function.
(taken from https://medium.com/data-science-bootcamp/understand-the-softmax-function-in-minutes-f3a59641e86d).

models of Fastai library with our data set, ResNet-34 gave 59% accuracy and DenseNet-201 gave 55% accuracy. The fact that the DenseNet-201 model's epochs lasted longer and gave a lower accuracy rate led us to define our next models as ResNet-34.

Then, as shown in the graph, we concluded that the failure to distribute the data properly affects the accuracy of the model. So we found a new data set. We combined these two data sets and tried to train the resnet34 model. The accuracy of this model was 74%. However, we still did not achieve the desired accuracy. We thought this was related to the low resolution of the images in the data set. Because the size of the pictures was 48x48.

We thought to increase our data by using animation data. Because the emotions of the images in our data set were not clear. Before using our animation data set, we made some preliminary operations. The animation dataset contained 768 * 768 images. First of all, we made face detection using OpenCV. After that, we resized the data set we

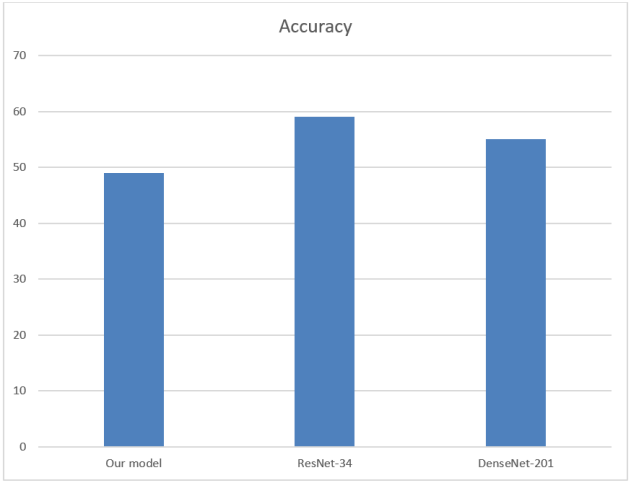

Figure 7. Accuracy for one data set.

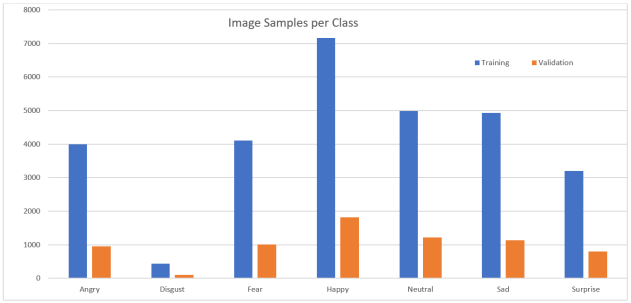

Figure 8. Distribution of classes for the first data set.

obtained as 48 * 48 images.

Due to the large size of the data set, the single epoch took 8 hours to train the model in Colab. Although it was the only epoch, the accuracy rate was 84%. This confirmed our belief that the problem was based on the data set. But since the training would take so long, this was not a model that could be improved for us. In addition, using animation data for training, validation, and testing did not precisely measure the efficiency of animation data for humans. That's why we regularly distributed 4.2k animation data to the training part of our previous 2 human data.

As a result, when we put some of the animation data set in the training part of the human data set, the ResNet-34 model gave 81% accuracy. Thus, we realized that the animation data had an effect on this model. The accuracy of our model can be increased by adding more animation data.

Besides, we also wanted to try our model in real-time. For this, we tested our model instantly using web-cam. We kept the snapshots in the video as frames continuously. For the images to be tested successfully, only the face in the image had to be detected. Therefore, we used the OpenCV library for face detection. So we identified the faces in the

Model	Number of Dataset	Accuracy
ResNet-34	2 human dataset	%74
ResNet-34	2 human dataset and all animation data (only one epoch)	%84
ResNet-34	2 human dataset and a part of animation data	%81

Figure 9. Accuracy for all data set.

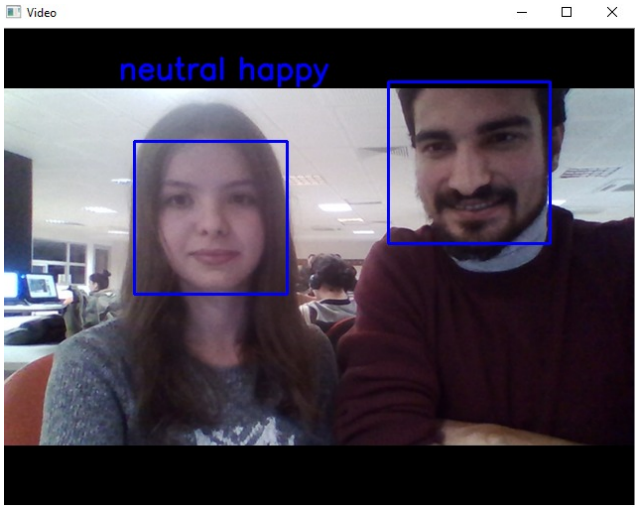

Figure 10. Real-time testing.

video and tested them with the model we created. In addition to determining the emotion of a person in the video, we set it up to determine the emotion of all faces identified in the video. In fact, we have proved the usability of our project. Thus, we have prepared an infrastructure to be used in all areas of use mentioned earlier.

5. Conclusions

As stated above our project may be used a great variety of areas. People do not always express themselves orally, but they always state something in their facial expressions. If we can know what they feel, then we can provide them a life that is more peaceful , easier and safer. Hence, we wanted to benefit from the very popular and powerful machine learning algorithms. By using these algorithms, we detect people's faces in a flowing video in real-time or in an image, find the prominent areas in their faces, and infer meaningful results from the angle between these areas. These are what our model basically does.

As step-by-step key results of our project, firstly we tried to create our own model. We used our first and unbalanced data set. Our accuracy was under what we expect from our project. Then we research new data set and combined two data sets. We also tried to use ResNet-34 and DenseNet-201 model for our project. After making some pre-processing data set and using these models, our accuracy improved se-

riously. After these processes, we learn that animation data is very useful to train our model more accurately. Then, we added 4.2k animation images to our only train set. This improved our accuracy to 81

We think about what we can do for our project in the future. In addition to projects such as increasing the driver's focus with the suggestion of music mentioned or improving the efficiency of speech by taking a break in the previous chapters, we encountered microexpressions in our investigations. A microexpression is the innate result of a voluntary and an involuntary emotional response occurring simultaneously and conflicting with one another [7]. We found this idea very exciting. This future work may blow the gaff in interrogation rooms.

References

[1] James, J. (1999). Beden dili. (Çev: M. Sağlam). İstanbul: Alfa Kitabevleri.

[2] Gamble, T. and Gamble M. (1990). Communication works. New York: Mc Graw Hill Publishing Company.

[3] https://www.kaggle.com/jonathanoheix/face-expression-recognition-dataset10053.jpg

[4] https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data

[5] https://grail.cs.washington.edu/projects/deepexpr/ferg-db.html

[6] https://www.geeksforgeeks.org/activation-functions-neural-networks/

[7] https://en.wikipedia.org/wiki/Microexpression