Lecture evaluate system based on emotion recognition
Előadás értékelő rendszer érzelemdetektálási alapokon
(revised December 2019)

Kerekes Ákos, Gaják Tibor István, Hannos Attila

Abstract—Nowadays the solutions with Artifical Intelligence has been gaining more and more ground. Knowledge of the solutions can be used in almost every field. One of the large areas, where the development is significant, is the field of image processing. It demonstrates the universality of the topic, that the solutions can be used in medieval field, where skin lesions detected by image processing, or the benefits utilize in the field of self-driving cars. Facial or emotional recognition is rapidly developing within image processing. Mostly the social media industry uses facial recognition, where the applications use a wide range of filters. However the field of facial and emotion recognition is untapped, so developers have many possibilities. In this semester, we have developed an educational assistant application, which concludes the quality of the lecture with the help of emotion detection. So the students are not required to fill tests as feedback. In the future we can develop our app further. Our idea, is to use this app in the financial sector, to examine the services' quality.

Kivonat—Manapság a XXI. század egyik legjobban felkapott témája a Mesterséges Intelligencia és azon belül a Deep Learning alkalmazása. Számos területen alkalmazzák a témában elért eredményeket. A dolgozat célja egy képfeldolgozás témakörében megoldandó probléma felvetése majd megoldása Deep Learning eszközök segítségével. Az emberek gyakran nem tudják és nem is akarják eltitkolni érzéseiket. Ezek az érzések kiülnek az arcukra. Célunk egy olyan rendszer fejlesztése ami ezeket az érzelmeket képes detektálni. Az információ bírtokában rengetek alkalmazásmódot el lehet képzelni. Gondoljunk csak bele, hogy egy banknak, vagy akár bármilyen szolgáltatónak az az információ, hogy a szolgáltatást igénybe véve az ügyfél elégedett volt-e vagy szomorúan esetleg mérgesen távozott, az nagyon sokat tud segíteni a szolgáltatás minőségének javításában. Szintén egy alkalmazási lehetőség egy tanárnak felmérni, hogy a diákok hogy érezték magukat az előadáson, milyen érzésekkel távoztak. A fejlesztendő alkalmazás érzelmeket detektál a képeken majd összesíti a talált érzelmeket és megjeleníti kördiagram formájában illetve jellemzi az adott szolgáltatást/előadást egy összesített érzelemmel.

1 Introduction

In the area of image processing, learning algorithms are often used. Think about self-driving cars. The algorithms of conventional vision are beginning to fall into the background. We have got acquainted with deep neural networks for image processing, and have used them in a concrete chosen topic. As for emotion and facial recognition there are many realizations, but as for the utilization of recognized emotion, there can be uncounted numbers of application ideas. Our idea was to create a system that could help the teacher by providing a feedback to the teacher based on the detected emotions, looking at the students faces and emotions, so the teacher can more easily decide that the students are satisfied with the lesson, or rather bored, angry or sad.

2 Previous solutions

2.1 First article we found about facial recognition

This solution proposes a new framework for facial expression recognition using an attentional convolutional network. They believe attention is an important piece for detecting facial expressions, which can enable neural networks with less than ten layers to compete with much deeper networks for emotion recognition. They also deployed a visualization method to highlight the salient regions of face images which are the most crucial parts thereof in detecting different facial expressions [1].

2.2 Second article

The second solution proposes a deep Emotion-Conditional Adaptation Network (ECAN), which conducts unsupervised cross-database facial expression recognition. The strength of ECAN lies in its ability to make the most of the beneficial knowledge from the target domain to simultaneously bridge the discrepancy of both marginal and conditional distribution between source and target domains, and also the discriminative power brought about by thre learning process of deep network. Besides, class imbalance problem has been taken into account and re-weighting parameter is introduced to balance the class bias between source and target domains. All these optimal goals associate with each other and then effectively boosts the recognition rate of cross-database facial expression recognition. Extensive experiments on widely-used facial expression datasets show that the proposed ECAN achieves excellent performance in a series of transfer tasks and outperforms previous cross-dataset facial expression recognition results, demonstrating the effectiveness of the proposed method [4].

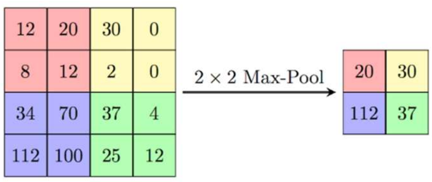
2.3 Third article

They designed a new and efficient CNN model for facial feature extraction, and proposed a complexity perception classification(CPC) algorithm for facial expression recognition. Our algorithm distinguished easily classifiable subspace from difficultly classifiable subspace to achieve discriminant learning of the facial feature distribution. Experimental results on the Fer2013 and CK+ datasets demonstrated that our algorithm outperformed state-of-the-art methods for facial expression recognition in terms of mean recognition accuracy [6].

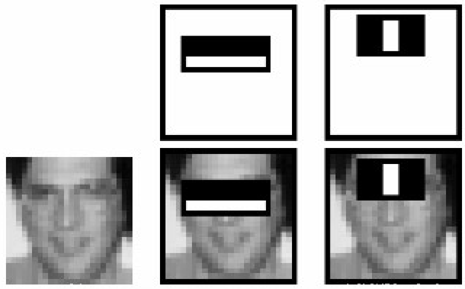
3 System design

3.1 Structure of a deep neural network

As for emotion detection we use convolutional neural networks, as is commonly used in image processing. We add max-pool layers to the network. The reason for this is that if we place deeper convolutional layers one after another, it can be seen that after a while the size of the activation tomb increases enormously. Therefore, after some layers we should perform some pooling operations per layer, which will reduce the size of the activation tomb We used the max-pool procedure. The operation of the procedure is illustrated in the following figure.

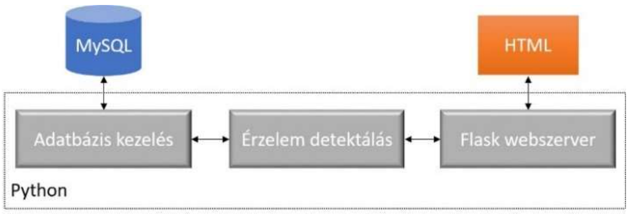


In addition to convolutional layers and max-pool layers, the network also includes dropout layers. The essence of this type of layer is that during the training stap, at each propagation step, a fraction of the activations of each layer are randomly reset (this effectively disconnects parts of our network). We need these layers to reduce the possibility of overfitting. Of course, we also need batch normalization to avoid overfitting and to improve convergence. The batch normalization technology means that continuously calculating the averages and standard deviations of mini-batches to be processed simultaneously, each activation is continuously normalized during training. Towards the end of the network, we also need fully connected layers, and the output of the network is such a layer with the appropriate number of neuronal and soft-max activations to generate probability variables. On the other hand, the important thing is that our network only works well if we get one face at the input not a big image with multiple faces of many objects. For sorting the faces in case of image processing we used the OpenCV directory.Cascade Classifier Haar cascade function is used. This procedure is usually used for face detection. The method uses a special kind of image feature called Haar image feature. These features examine a portion of image by means of a binary window, the output of which is large positive number for similarity and a large negative number for discrepancy, about zero if there is no similarity.



3.2 Structure of the application

The soul of the system is provided by the emotion detection module, which was designed as a large class. For our application, we used thread management to serve the web interface. At start of the application instantiates the emotion detection class and then launches the web interface and the emotion detection on a separate thread on the connected camera. Connecting to the MySQL-based database is part of the startup process. By connecting to the database, we can save various detected emotions and compile statistics.



4 Realization of the application

4.1 Collecting datas:

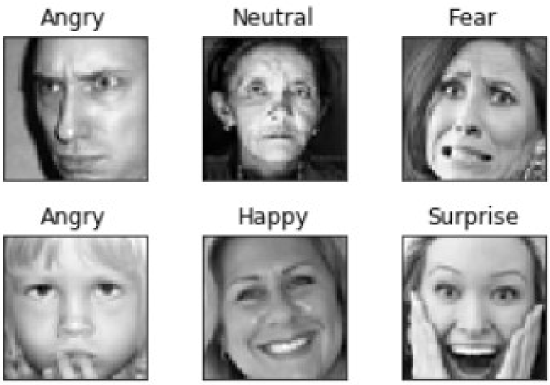
The first step is creating or searching the database. Database creation is one of the most costly parts of this field, especially for supervised teaching where you need to associate the required label with the training data. While browsing public databases, we found the fer2013 database

	emotion	pixels	Usage
0	0	70 80 82 72 58 58 60 63 54 58 60 48 89 115 121...	Training
1	0	151 150 147 155 148 133 111 140 170 174 182 15...	Training
2	2	231 212 156 164 174 138 161 173 182 200 106 38...	Training
3	4	24 32 36 30 32 23 19 20 30 41 21 22 32 34 21 1...	Training
4	6	4 0 0 0 0 0 0 0 0 0 0 0 0 3 15 23 28 48 50 58 84...	Training

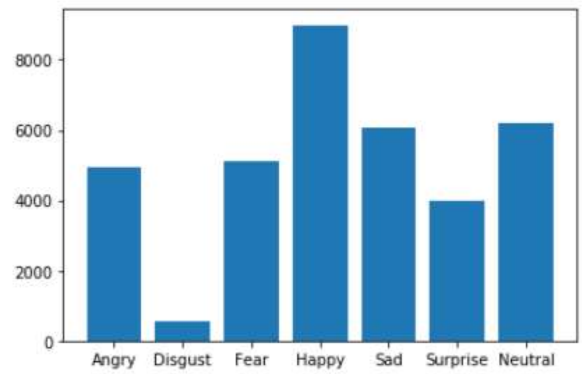
on the Kaggle website, which contains 48x48 pixels of faces in a csv file format. The database distinguishes seven emotions. Emotion with zero value is angry, value one is disgust, number two is fear, number three is happy, number four is sad, number five is surprise, number six is neutral.

4.2 The structure of the database:

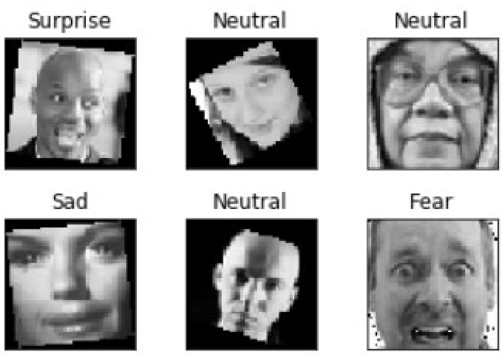
The first step was to know and visualize the data in the database.



After visualization, we prepared the data for teaching, but examining the distribution of the datas across the groups showed that the distribution was not equal.



The images were artificially transformed for greater accuracy, this process is the image augmentation, thus balanc-



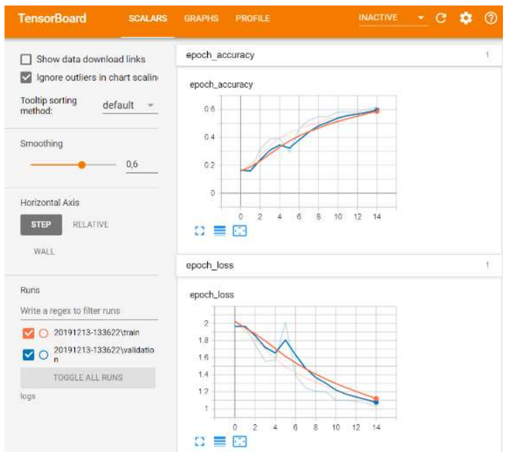
ing the number of images in different. Such augmentation techniques include, for example, mirroring the image, rotating it slightly, and making noise.

With the help of the train_test_split function, which is the part of the sklearn library we splitted the new augmented database for three section: 70% train, 20% validation, 10% train.

4.3 The process of the training

We used the Keras framework for training. After the creation of the database, we had to define our model, and the optimalization method. We used different callback functions to make our model training more accurate. It was like the ReduceLROnPlateau Keras function, which has an effect on the learning rate when a metric has stopped improving. The next callback functions were the ModelCheckpoint and the EarlyStopping. These functions should be used together. With the help of ModelCheckpoint, we can save our best model. The EarlyStopping is used against the overfitting. The running length of a train is also a hyperparameter If we don't use enough epochs for train, the model doesn't train in the desired way to the test dataset.

If we overrun the training, which means the model is overtrained, so in the test dataset we can reach 100% accuracy, but the expanse of generalization, the network will begin to memorize the teaching data, so it will be less accurate if a new image is detected. It is difficult to specify enough epochs, so we use EarlyStopping. The essence of the method is, when a failure qualifying metrics do not decrease (often the validation loss), the training will be stopped. It has a patience option, which tells how long we shouldwait with the constant error value to stop teaching. The last callback which we use is the TensorBoard. This supplementary function provides informations about the training in a web interface with the help of graphs. This can



be useful for example in optimizing different parameters.

4.4 Hyperparameter optimization

This method is used to optimize and improve the network. One deep neural network possesses plenty hyperparameters, for example: the learning-rate, which modifies the step-size in gradient-based methods, the numbers of the network secret layers, the numbers of neuron sin one layer, the filter size in convolutional layer, batch size, etc. The previous list shows how vast is the hyperparameter space. If we want to test every combination of the listed paramters, the algorithm will never be finished. As for the acceleration of the hyperparameter optimization we used different packages. With the help of these packages, we can randomly pair two parameters and epoch-by-epoch we can train the network with them.

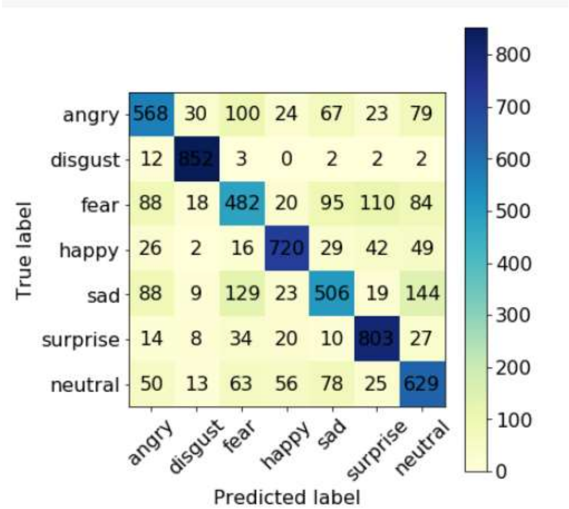
5 Evaluation

We retrained the model many times with different parameters. At first we did the hyperparameter optimization manually, but after that we automated the changes of net parameters epoch-by-epoch.

```
Epoch 00048: val_loss did not improve from 0.85071
Epoch 49/100
44045/44045 [==============================] - 151s 3ms/step
loss: 0.3953 - accuracy: 0.8690 - val_loss: 0.9024 - val_accuracy: 0.7290
Epoch 00049: val_loss did not improve from 0.85071
Epoch 00049: early stopping
<keras.callbacks.callbacks.History at 0x23ee3b25588>
```

In the picture our best model can be seen. The validation loss was reduced to about 0.9, and the validation accuracy was 73%.

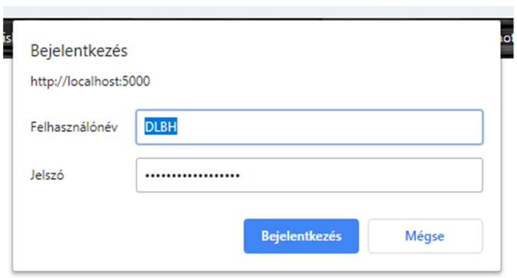
We used the confusion matrix to evaluate our model.

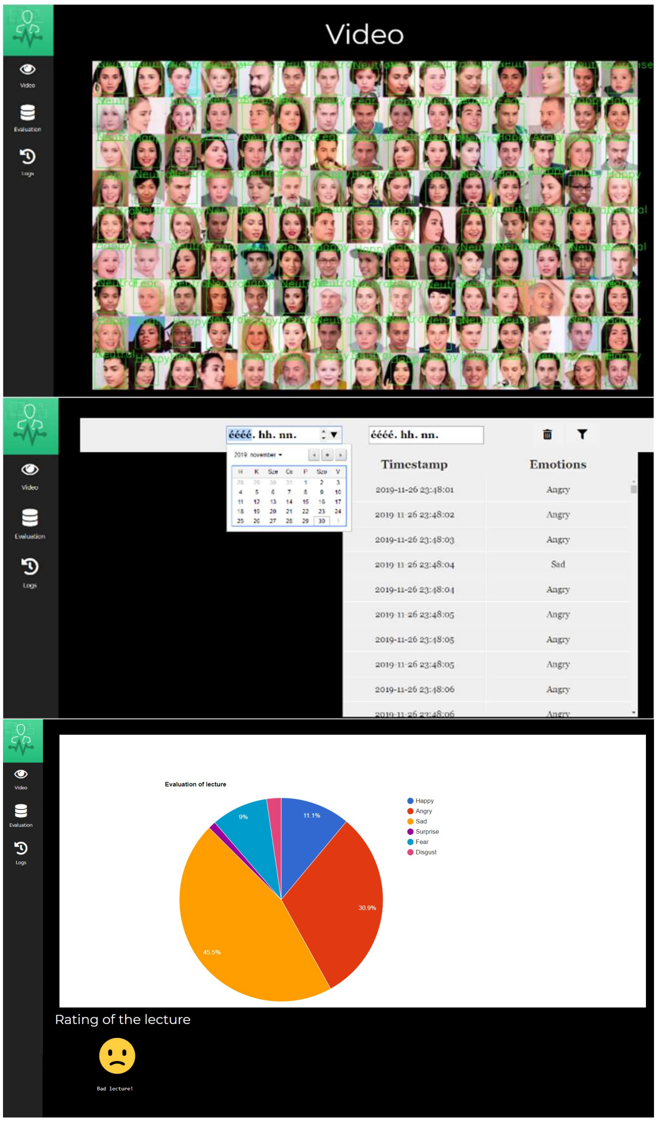


With the help of the confusion matrix, we can see how our model identifies the pictures in the test dataset, where makes the most mistakes, and where is the most accurate. The matrix shows us, mostly the pictures with sad emotion identified wrongly as neutral emotion, however the detection of disgust emotion is the most accurate. The accuracy of our model, in the test dataset was around 64%.

6 Introduction of the Application

The final version of the application is able to analize videos or camera pictures in a web interface. It is capable of displaying the detected faces and the adherent emotions side-by-side. The app stores the detected emotions in a database. The user can filter the database according to time, and also there is a diagram, which summarizes all the different emotions. The algorithm uses the summarized datas, and qualifies the lecture.





7 Conclusion

Through solving our task, we examined the basic methods of Deep Learning. We successfully trained a deep neural network, so we expanded our engineering knowledge. We faced interesting problems, such as the areas vast resource requirements. Algorithms, which run for days, weeks we did not meet.

8 Future Planes

Our application can be improved in many ways. The first option is the possibility to upload and analyze the picture. Also we would like to test different technics, layers, net constructions to improve the accuracy of our model.

REFERENCES

[1] Shervin Minaee, Amirali Abdolrashidi Expedia Group University of California, Riverside Deep-Emotion: Facial Expression Recognition Using Attentional Convolutional Network https://arxiv.org/pdf/1902.01019.pdf

[2] Ching-Da Wu and Li-Heng Chen The University of Texas at Austin FACIAL EMOTION RECOGNITION USING DEEP LEARNING https://arxiv.org/pdf/1910.11113.pdf

[3] Suraj Tripathi1, Abhay Kumar1, Abhiram Ramesh1, Chirag Singh1, Promod Yenigalla1 1Samsung R&D Institute India – Bangalore Deep Learning based Emotion Recognition System Using Speech Features and Transcriptions https://arxiv.org/ftp/arxiv/papers/1906/1906.05681.pdf

[4] Shan Li, and Weihong Deng∗ , Member, IEEE A Deeper Look at Facial Expression Dataset Bias https://arxiv.org/pdf/1904.11150.pdf

[5] Shima Alizadeh, Azar Fazel Convolutional Neural Networks for Facial Expression Recognition https://arxiv.org/pdf/1704.06756.pdf

[6] Tianyuan Chang, Guihua Wen, Yang Hu, JiaJiong Ma Facial Expression Recognition Based on Complexity Perception Classification Algorithm https://arxiv.org/ftp/arxiv/papers/1803/1803.00185.pdf

[7] Ram Krishna Pandey, Member IEEE, Souvik Karmakar, A G Ramakrishnan, Senior Member IEEE and N Saha Improving Facial Emotion Recognition Systems Using Gradient and Laplacian Images https://arxiv.org/pdf/1902.05411.pdf

[8] Samanyou Garg GROUP EMOTION RECOGNITION USING MACHINE LEARNING https://arxiv.org/ftp/arxiv/papers/1905/1905.01118.pdf

[9] Facial Emotion Recognition using Convolutional Neural Networks https://arxiv.org/pdf/1910.05602.pdf

[10] Predicting Personal Traits from Facial Images using Convolutional Neural Networks Augmented with Facial Landmark Information https://arxiv.org/pdf/1605.09062.pdf

[11] DeXpression: Deep Convolutional Neural Network for Expression Recognition https://arxiv.org/pdf/1509.05371.pdf