Analysing Facial Expressions for Emotion Recognition using Machine Learning Algorithms

Gonçalo Matos
Tópicos de Aprendizagem Automática 2020/2021
Departamento de Eletrónica, Telecomunicações e Informática
Universidade de Aveiro
Aveiro, Portugal
gmatos.ferreira@ua.pt

Margarida Martins
Tópicos de Aprendizagem Automática 2020/2021
Departamento de Eletrónica, Telecomunicações e Informática
Universidade de Aveiro
Aveiro, Portugal
margarida.martins@ua.pt

***Abstract*—Analysing facial expressions is often used in emotion recognition. The aim of this paper is to analyse the performance of machine learning deep learning models, that classify emotions through facial images taken in the wild. For this purpose 9600 images were used from the AffectNet dataset [1]. Using Jupyter notebooks, two different models were tested, Convolutional Neural Networks with different complexities. The best model achieved an accuracy of 35,19% classifying 8 different emotions.**

***Index Terms*—emotions recognition, machine learning, deep learning, convolutional neural networks, facial expressions, keras, python, jupyter notebooks**

I. Introduction

Following the last project where we have explored machine learning algorithms for supervised learning (Artificial Neural Networks and Support Vector Machines), in this second one we were proposed to explore a more advanced technique. We have implemented several variations of the deep learning model convolutional neural networks (CNN).

To better understand how this models perform in relation to the explored previously and because we believe this is a pertinent subject, we have kept the topic: analysing facial expressions.

To implement both we have recurred to Keras [2] library for Python, a high-level deep learning library that simplifies the process of creating and training models. It is part of TensorFlow API.

II. State of the art

Before starting our project we have searched for papers handling problems similar to ours. In this chapter we explore those we found more suitable.

One paper [3] published in *IEEE Access electonic journal* has proposed a work with 88.56% accuracy for distinguishing 4 emotions using CNN. This approach starts by locating the faces, normalizing their scale to 128x128 and converting them to grayscale with Histogram Equalization to adjust the contrast. The network architecture, built with Keras Python library, has 7 layers that result of a combination of three types: convolutional, polling and connective. In the end, the data goes through one Softmax layer classifier.

Another paper [4] published in the *2020 Chinese Automation Congress (CAC)* used CNNs in order to classify facial expressions divided in 9 levels according to the Valence-Arousal dimensional emotional model. The model was built using Keras and consisted of a CNN network with 4 convolution layers, 3 pooling layers and 3 fully connected layers. The data used for training the model was gray scaled images with a size of 48x48. Images needed preprocessing where the face area was detected and the image cropped in order to remove background noise. This model was tested by analyzing the facial expressions of volunteers while they watched small video clips which were divided in 3 different feelings: positive, neutral and negative.

In [5] , Sang, Cuong and Ha, a deep CNN model (DenseNet) is proposed in order to classify seven different emotions. The model architecture consists of 1 convolutional layer, three dense blocks and three transition blocks. With this model they were able to achieve an accuracy above 70%.

Faisal Ghaffar [6] has reached 78% accuracy by combining two datasets of 7 emotions. His work implements a CNN architecture with three convolutional layers, each followed by a pooling one and then three dense layers. The image preprocessing has manipulated several aspects, starting with the face detection, extraction, resize (to 100 x 100) and histogram equalization. The face detection was achieved using a pretrained model to identify the 68 facial landmarks. Because of the reduced number of samples, each image was duplicated five times, each with a different filter applied.

In the CNN area but with a different approach we found the *University of California* paper [7] that follows a deep learning approach, but using attentional convulutional network. This method is a variation of the ”traditional” CNN and focuses on important parts of the face, which allows promising results with less that 10 layers. It is implemented through a localization network, made up of two convolutional layers (each followed by a max-pooling and a RELU) and two fully-connected layers. The results mention experiments on several datasets, some with images taken in wild settings, in controlled environments and even one with stylized characters. The accuracy varies from 99.3% in the best dataset and 70.02% in the worst.

In the cs231 Stanford course a pair of students developed CNNs for facial expression recognition of seven different emotions [8]. They compared two different architectures. The first one is a shallow CNN with 2 convolutional layers and 1 fully connected layer. The second one is deeper with 4 convolutional layers and 2 fully connected layers. The accuracies obtained were 49% for the shallow model and 61% for the deeper model, meaning that the problem at hands is a complex one.

III. Data description, visualization and statistical analysis

In this project the mini version of AffectNet [1] database was used. This database consists of 420299 images divided in train and validation sets. The images have a size of 224*224 pixels (RGB color) and are classified into eight different emotions: anger, contempt, disgust, fear, happy, neutral, sad and surprise as we can see in Figure 1 bellow. Due to training time issues only 9600 samples were retrieved from the AffectNet training set to be used as training data, which corresponds to 1200 examples per emotion. The test images consisted of 200 images per emotion retrieved from the AffectNet validation set.

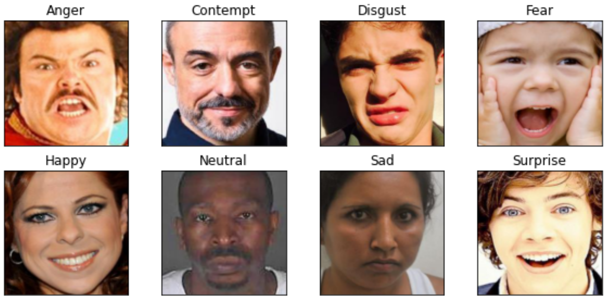


Fig. 1. Sample images from the dataset

Along with the images the database provided annotations for each one. The Arousal/Valence values were given as well as 68 facial landmarks points. In Figure 2 we can see an example of the landmarks annotated for a sample image.

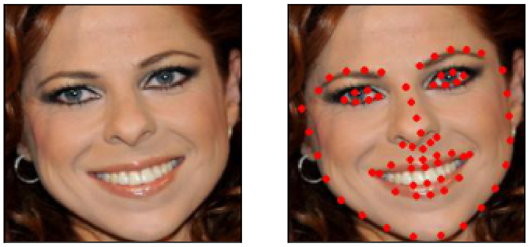


Fig. 2. Facial landmarks annotated for a sample image

IV. Data preprocessing

In order to remove as much background noise as possible, all images were cropped using the annotated landmarks. Then all the images were reshaped to a 48*48 pixel size. This way we were able to have the faces in the photos centered and occupying the same amount of space.

Given the problem context we considered color unnecessary and converted all the images to gray scale in order to reduce the image size, instead of 3 we have only 1 color channel. As also mentioned in Chapter II, histogram equalization was used in order to enhance the images contrast. This technique works very well in gray scale images and consists in spreading out the most frequent intensity values of a picture. Lastly we normalized our data, dividing all images by 255 in order to only have values between 0 and 1 for all the images.

Figure 3 allows for a better visualisation of the data preprocessing.

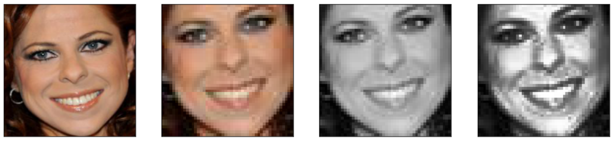


Fig. 3. Example of data preprocessing

A seemingly simple classification problem gave us 2304 (48*48*1) non-linearly separable features per example to work with, a number too high for logistic regression. We had to consider more complex approaches.

A. Computer vision

This scientific field has been studying ways to create computational models of human visual system in order to mimic its behaviour and automate it [9], having already proposed several of them. In our lessons we've learned two and we are going to explore them in this paper: Neural Networks (NN) and Support Vector Machine (SVM).

B. Data splitting

Due to the large number of samples in the dataset we were advised by our teacher to train the network only with 1000 examples per emotion. We also created development and testing sets with 200 images each.

The development images were retrieved from the test set because the original division did not have one and development and test examples should come from the same source, so as to improve accuracy values. The original test set had more than enough examples so no images had to be repeated.

To analyse the relation of the accuracy variation with the number of emotions we used to train our model and to compare the result of this model with the Neural Network and Support Vector Machine used in the first project, we have also trained it for the 6 emotions analysed in the project mentioned: Angry, Fear, Happy, Neutral, Sad and Surprise.

V. Classes' Convolutional Neural Network

In classes we have explored a simple model which purpose was to distinguish two possible facial states given photos of faces: happy or not happy. With 3 convolutional layers we have obtained a very high accuracy.

A. Parameters definition

Due to the complexity of our problem we have studied how the the variation of convulutional layers number, epochs and regularization affected the model.

For the class problem we have reached the better result with 3 layers, 5 epochs. For our problem we have chosen to variate the number of layers between 2 and 4 and to start with 20 epochs.

When we trained the first model, however, we have noticed that after the 4th epoch, as seen in Fig 4, the metrics start to diverge, with the loss function increasing and the accuracy of the training set rising with the validation one stable, which indicates overfitting. We have then chosen to train our model with 4 epochs.

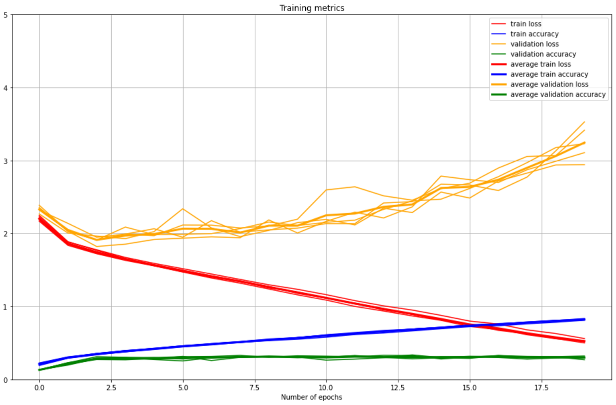


Fig. 4. Training metrics for 3 layers with 20 epochs

For the regularization we have explored two values for gama: 0.1 and 3.

B. Training

This model consists of a Keras sequential model with the already explained variable number of convolutional layers and an output (Dense) layer.

For training purposes we have also used:

1) Activation functions: To facilitate the parameter optimization, we implemented Relu activation function on the hidden layers and Softmax for the final layer, as it is widely used for classification with multiple labels.

2) Cost function: The cost function we used is the sparse categorical crossentropy loss function, also widely used in classification problems.

3) Optimizer: To optimize the model against the cost function and ensure it converges to an optimal solution, we have used the stochastic gradient descent Adam algorithm.

In addition, we have applied K-fold Cross Validation on the training set, dividing it in 5 subsets, in order to estimate the skill of our model.

C. Results

The K-fold results are pretty similar for each subset, with each evolving the same way as the others. Fig. 5 show the metrics evolution for the loss and accuracy for every epoch number for 2 and 4 number of layers.

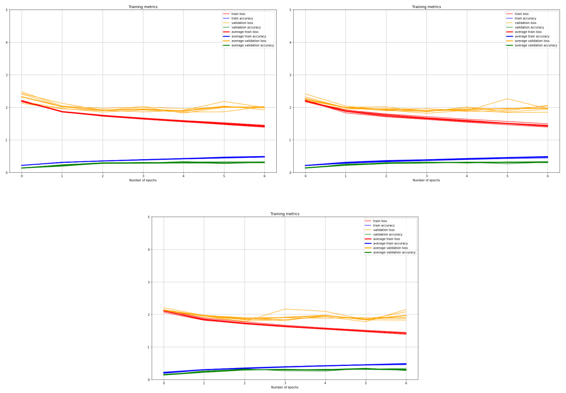


Fig. 5. Training metrics evolution with epochs variation for classes' CNN with (a) 2 (b) 3 and (c) 4 layers

When analysing the regularization parameter, the accuracy evolution is also similar for the different number of CNN layers. As visible in Fig. 6 below, when the parameter is set to 3 the accuracy does not change, being always around 12.5%, which indicates underfitting. For a smaller parameter, 0.1, the evolution is evident, increasing with the epochs and stabilizing after the 3rd on values around 30%.

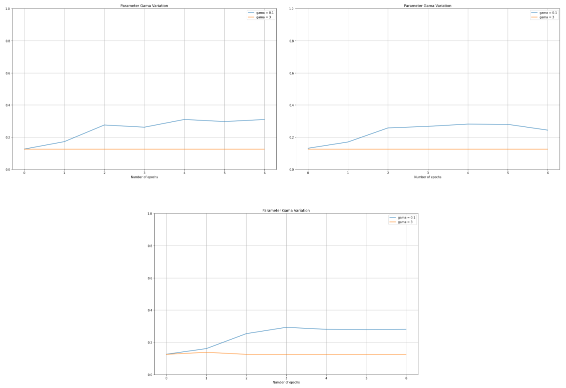
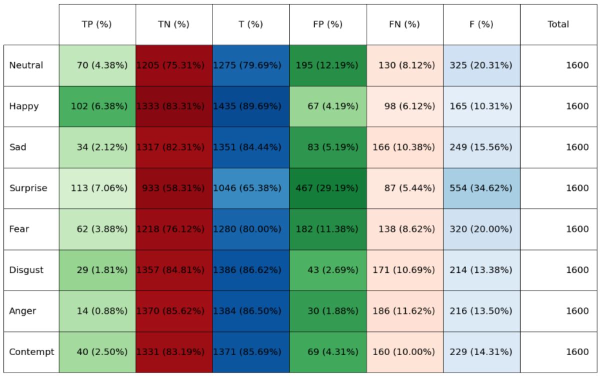


Fig. 6. Accuracy evolution with gama variation for classes' CNN with (a) 2 (b) 3 and (b) 4 layers

Discovered the best parameters, we have retrained the model with it: 4 epochs and a regularization parameter of 0.1. As for the previous variations, the accuracy did not change for the different number of CNN layers, returning 28.5%, 31.5% and 29% for the test set with 2, 3 and 4 layers.

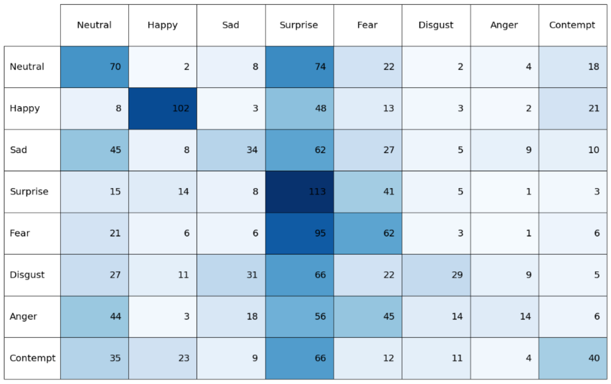
As the results are very similar, further we are going to explore only one of this variations: the model with 3 layers (which performed best). The accuracy stated before already suggested a low performance with the test set. In Fig. 7 we can see that most of the positive classifications were false positives, having happy been the only one with a greater rate of true positives. On the opposite side is surprise, with the greatest number of false positives.



	TP (%)	TN (%)	T (%)	FP (%)	FN (%)	F (%)	Total
Neutral	70 (4.38%)	1205 (75.31%)	1275 (79.69%)	195 (12.19%)	130 (8.12%)	325 (20.31%)	1600
Happy	102 (6.38%)	1333 (83.31%)	1435 (89.69%)	67 (4.19%)	98 (6.12%)	165 (10.31%)	1600
Sad	34 (2.12%)	1317 (82.31%)	1351 (84.44%)	83 (5.19%)	166 (10.38%)	249 (15.56%)	1600
Surprise	113 (7.06%)	933 (58.31%)	1046 (65.38%)	467 (29.19%)	87 (5.44%)	554 (34.62%)	1600
Fear	62 (3.88%)	1218 (76.12%)	1280 (80.00%)	182 (11.38%)	138 (8.62%)	320 (20.00%)	1600
Disgust	29 (1.81%)	1357 (84.81%)	1386 (86.62%)	43 (2.69%)	171 (10.69%)	214 (13.38%)	1600
Anger	14 (0.88%)	1370 (85.62%)	1384 (86.50%)	30 (1.88%)	186 (11.62%)	216 (13.50%)	1600
Contempt	40 (2.50%)	1331 (83.19%)	1371 (85.69%)	69 (4.31%)	160 (10.00%)	229 (14.31%)	1600

Fig. 7. Test set performance for model with λ=0.1 and 3 CNN layers

The confusion matrix helps us making a more detailed analysis. Fig 8 shows that the model has a high precision (60.4%) when classifying emotion happy. This contrasts with every other emotion, having been classified more as other than itself, with highlight again for surprise, highly misclassified, with a precision of 19.5%.



	Neutral	Happy	Sad	Surprise	Fear	Disgust	Anger	Contempt
Neutral	70	2	8	74	22	2	4	18
Happy	8	102	3	48	13	3	2	21
Sad	45	8	34	62	27	5	9	10
Surprise	15	14	8	113	41	5	1	3
Fear	21	6	6	95	62	3	1	6
Disgust	27	11	31	66	22	29	9	5
Anger	44	3	18	56	45	14	14	6
Contempt	35	23	9	66	12	11	4	40

Fig. 8. Test set confusion matrix for model with λ=0.1 and 3 CNN layers

VI. More complex Model

Based on a model created for emotion recognition 9 we developed a new model, more complex than the one used in class. The model's architecture is described in the figure below. It consists on 8 convolutional layers, 4 pooling layers and 2 dense layers.

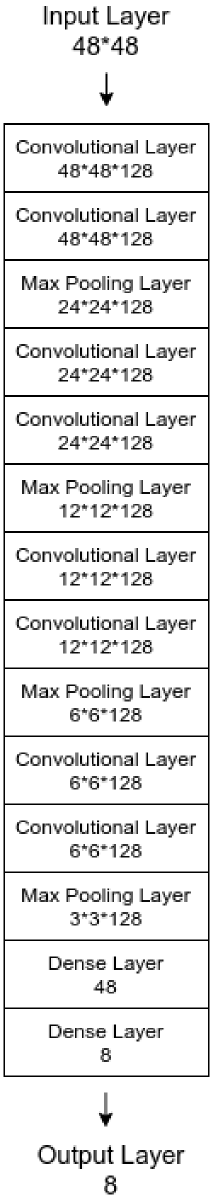


Fig. 9. Model Architecture

For the convolutional layers it was used the relu activation function and filters with a 3*3 dimension, all convolutional layers have the same number of filters. Regarding the pooling layers it was used a max-pooling approach with a stride size of 2*2 and a pool size of 2*2. The first dense layer has relu as activation function while the second one uses softmax because is the last layer and this is a classification problem with more than two classes. The optimizer used was Adam.

A. *Parameters definition*

Our model has 3,321,560 trainable parameters (considering convolution layers with 128 filters) with a total of 3,324,344 parameters. Given time constraints, we fixed some parameters such as the convolutional filter dimension, the pooling size and stride. Other parameters were defined using K-fold cross-validation. The values experimented were as follows:

- Number of convolutional filters: 32, 64, 128;
- Hyper parameter: 0, 0.001, 0.01, 0.1, 1;
- Learning rate: 0.0001, 0.001, 0.01;

B. Training

For each parameter combination the model was trained using k-fold cross validation with 5 folds with the accuracy being calculated based on the average of each fold prediction using the validation set, an example of this method results can be seen in figure 10.

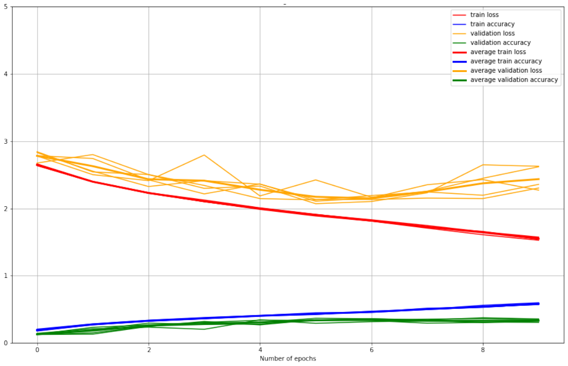


Fig. 10. Example of k-fold cross validation results

With the best parameters found, the model was trained again using the complete training and validation set as training data. The final confusion matrix and metrics are calculated using the test set.

C. Results

The first parameter analyzed was the number of filters in the convolutional layers. Figure 11 shows the validation accuracy for the different values experimented. As we can see the parameter value with best validation accuracy is 128. 64 or 32 filters are not complex enough for our problem. After this experience a model with 256 number of convolutional filters was also tested but the validation accuracy was lower than the other three values as it overfitted.

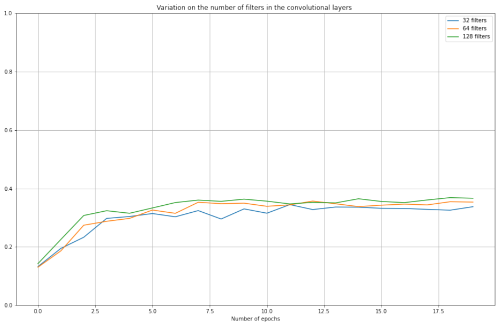


Fig. 11. Validation accuracy with multiple number of convolutional filters

After setting the number of convolutional filters, variations of the gamma value were tested as we can see in figure 12. The model does not learn when the gamma value is equal to one. As gamma decreases the validation accuracy increases. Therefore gamma was fixed as 0.

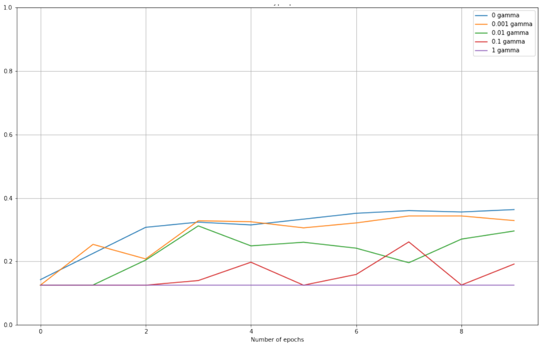


Fig. 12. Validation accuracy with multiple gamma values

The last tuned parameter was the learning rate. The best learning rate value was 0.001, closely followed by 0.0001 as shown in figure 13. However an higher learning rate value such as 0.01 causes the cost function to diverge resulting on low validation and training accuracies, since the model is not able to learn.

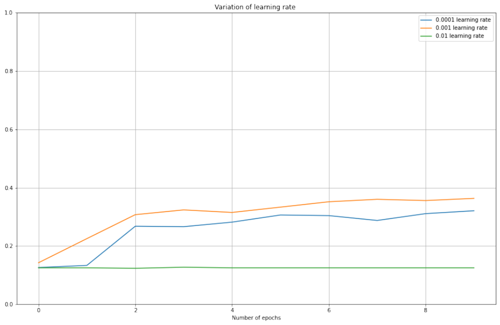
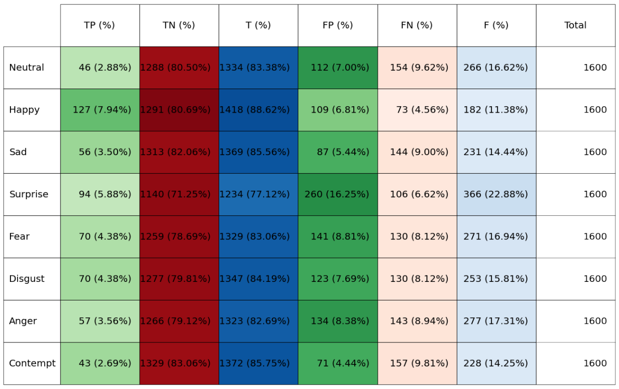


Fig. 13. Validation accuracy with multiple learning rate values

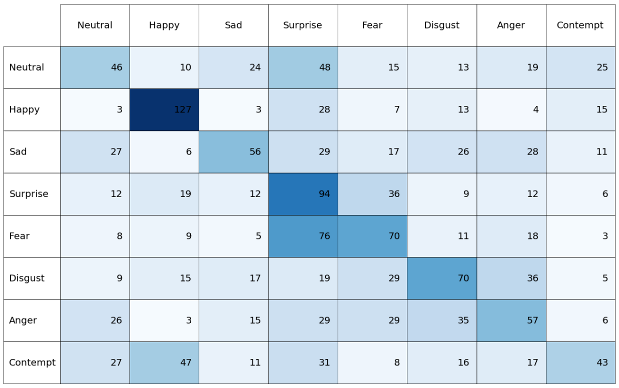
Having set all the parameters and trained the model, predictions were made with the test data. An accuracy of 35,19% was obtained. Given the fact that the number of emotions is balanced, the overall accuracy is a good metric. From figure 14 it can be inferred that happy is the most accurately detected emotion (63,5%) while contempt is the least accurate (21,5%)



	TP (%)	TN (%)	T (%)	FP (%)	FN (%)	F (%)	Total
Neutral	46 (2.88%)	1288 (80.50%)	1334 (83.38%)	112 (7.00%)	154 (9.62%)	266 (16.62%)	1600
Happy	127 (7.94%)	1291 (80.69%)	1418 (88.62%)	109 (6.81%)	73 (4.56%)	182 (11.38%)	1600
Sad	56 (3.50%)	1313 (82.06%)	1369 (85.56%)	87 (5.44%)	144 (9.00%)	231 (14.44%)	1600
Surprise	94 (5.88%)	1140 (71.25%)	1234 (77.12%)	260 (16.25%)	106 (6.62%)	366 (22.88%)	1600
Fear	70 (4.38%)	1259 (78.69%)	1329 (83.06%)	141 (8.81%)	130 (8.12%)	271 (16.94%)	1600
Disgust	70 (4.38%)	1277 (79.81%)	1347 (84.19%)	123 (7.69%)	130 (8.12%)	253 (15.81%)	1600
Anger	57 (3.56%)	1266 (79.12%)	1323 (82.69%)	134 (8.38%)	143 (8.94%)	277 (17.31%)	1600
Contempt	43 (2.69%)	1329 (83.06%)	1372 (85.75%)	71 (4.44%)	157 (9.81%)	228 (14.25%)	1600

Fig. 14. Test set performance for best model

Figure 15 shows the model's confusion matrix.Fear is more likely to be classified as surprise than fear itself. On the other hand happy and anger are the emotions more distinguishable from one another with only 4 happy emotions classified as angry and 3 angry ones mistaken with happy.



	Neutral	Happy	Sad	Surprise	Fear	Disgust	Anger	Contempt
Neutral	46	10	24	48	15	13	19	25
Happy	3	127	3	28	7	13	4	15
Sad	27	6	56	29	17	26	28	11
Surprise	12	19	12	94	36	9	12	6
Fear	8	9	5	76	70	11	18	3
Disgust	9	15	17	19	29	70	36	5
Anger	26	3	15	29	29	35	57	6
Contempt	27	47	11	31	8	16	17	43

Fig. 15. Test set confusion matrix for best model

VII. Performance comparation between models

The "class model" presented has a lower accuracy value compared with the "complex model" (31,5% versus 35,19%). Some of the models' differences can be seen in table I. The "complex model" classifies the images as anger or disgust much more while the "class model" surpasses in over classifying surprise and neutral emotions. Both models classify images as surprise much more frequently than the number expected (200, the number of surprise examples in the test set). The difference between the emotion less and most classified is much bigger in the "class model" (536 versus 240).

	anger	fear	happy	neutral
Class Model	44	244	169	265
Complex Model	191	211	236	157

	sad	surprise	disgust	contempt
Class Model	117	580	72	109
Complex Model	143	354	193	114

TABLE I
Number of images classified by emotion

Looking at the models precision for each emotion represented in table II, we can conclude that the average precision is larger in the "complex model" (35.6% versus 33.7%) as expected. Both the best and worst emotion precision, happy and surprise respectively, belongs to the "class model". Surprise was imprecise in both models while happy being the best.

	neutral	happy	sad	surprise
Classes' CNN	26.4%	60.4%	29.1%	19.5%
Complex model	28%	53.8%	39.2%	26.6%

	fear	disgust	anger	contempt
Classes' CNN	25.4%	40.3%	31.8%	36.7%
Complex model	33.2%	36.3%	29.8%	37.7%

TABLE II
Precision by emotion

In our previous project, we used SVM and simple neural networks models to classify emotions. The SVM model was the best model with an accuracy of 38.8% when classifying 6 different emotions (anger, fear, happy, neutral, sad and surprise). In order to compare the performance of the "complex model" developed in this project with the SVM one a new model was trained with only this 6 emotions. The new model obtained has an accuracy of 42.83% a bit higher than the SVM model. Table III shows the differences between the models regarding the precision of each emotion. The average precision of the CNN model is higher (43.2% versus 38.3%). While in the SVM model surprise is the most precise emotion in the CNN model surprise precision is bellow average while happy's being the highest one. Both models have the same neutral precision.

	anger	fear	happy	neutral	sad	surprise
SVM	28%	28%	48%	34%	33%	59%
CNN	39%	42%	65%	34%	36%	43%

TABLE III
Comparation of precision by emotion

VIII. Conclusions

Despite the low accuracy obtained with the best model created, with this project we managed to improve the 38.8%

obtained with classical machine learning algorithms. With the several experiments we have improved our knowledge on deep learning algorithms and understood that complex problems require more complex approaches, combining several approaches and algorithms.

As in the previous project we conclude that emotions that are more clearly distinguished from others by humans are also better classified by our models, being happy the best one.

IX. Work Load

Each student worked 50% of the project.

Acknowledgment

The authors would like to thank professor Petia Georgieva, regent of TAA course, for her support, assistance throughout the project and the flexibility for the project delivery date.

References

[1] A. Mollahosseini, B. Hasani, and M. H. Mahoor, “Affectnet: A database for facial expression, valence, and arousal computing in the wild,” *IEEE Transactions on Affective Computing*, vol. PP, no. 99, pp. 1–1, 2017.

[2] “Keras website.” [Online]. Available: https://keras.io/

[3] H. Zhang, A. Jolfaei, and M. Alazab, “A face emotion recognition method using convolutional neural network and image edge computing,” *IEEE Access*, vol. 7, pp. 159 081–159 089, 2019.

[4] S. Liu, D. Li, Q. Gao, and Y. Song, “Facial emotion recognition based on cnn,” in *2020 Chinese Automation Congress (CAC)*, 2020, pp. 398–403.

[5] D. V. Sang, L. T. B. Cuong, and P. T. Ha, “Discriminative deep feature learning for facial emotion recognition,” in *2018 1st International Conference on Multimedia Analysis and Pattern Recognition (MAPR)*, 2018, pp. 1–6.

[6] F. Ghaffar, “Facial emotions recognition using convolutional neural net,” 2020.

[7] S. Minaee and A. Abdolrashidi, “Deep-emotion: Facial expression recognition using attentional convolutional network,” 2019.

[8] S. Alizadeh and A. Fazel, “Convolutional neural networks for facial expression recognition,” 2017.

[9] “Computer vision: Evolution and promise.” [Online]. Available: http://cds.cern.ch/record/400313/files/p21.pdf