Benchmark on networks to classified emotion from faces

Angelo Pili
S252524

Giovanni Zara
S252735

Abstract

Classification of customers and their interesses has achived large interess in the last past years. Most of e-commerce track customer researches to propose ad-hoc advertising. Our idea is to bring that aproach in the real word. Our system has the goal of classifying customers's facial reaction to estimate and study how much they like or dislike a product.

1. Introduction

Study the facial reaction could be a very important factor for understanding people and their opinion about the things that see during their daily routine. Unfortunately give a unique common classification about an expressed emotion is not realistic, therefore the goal of this kind of study is looking for a good approximation of this classification. This classification can be useful for various field in the for the real world for instance in the marketing for understand several interesting trends.

2. Related Works

That project is focused on train convolutional neural network. Some other groups did that in the past, for example Alexandru Savoiu and James Wong (Stanford University) in "*Recognizing Facial Expression Using Deep Learning*" perform a similar work where they study the performance of SVM, VGG-16 and ResNet50 and a combination of this architectures[1].

Another similar work that inspire us for this project is "*Touch Feely: An Emotion Recognition Challange*" from Dhruv Amin, Patrick Chase and Kirin Sinha (Stanford University) where they start from a simple baseline model and do experiments finding several networks for facial emotion recognition.[2]
These two papers inspire us to make this project.

3. Methods

For our project, we use several different architectures: AlexNet, GoogLeNet, ResNet50 and one model that we have implemented. For the existing architectures, we start with pre-trained ImageNet weights.

3.1. AlexNet

AlexNet represents one of the state of the art architectures for convolutional neural networks. It has 60 million parameters and 650,000 neurons.
AlexNet consists of 5 Convolutional Layers and 3 Fully Connected Layers. The first two Convolutional layers are followed by the Overlapping Max Pooling layers. The third, fourth and fifth convolutional layers are connected directly. The fifth convolutional layer is followed by an Overlapping Max Pooling layer, the output of which goes into a series of two fully connected layers. The second fully connected layer feeds into a softmax classifier with originally 1000 class labels. We have edited that Layer changing the output to 7 class labels. ReLU non linearity is applied after all the convolution and fully connected layers. The input to AlexNet is an RGB image of size 256×256. This means all images in the training set and all test images need to be of size 256×256.[3]

3.2. ResNet50

ResNet50 is another current state of the art convolutional neural network architecture.

It makes the improvement over AlexNet by replacing large kernel-sized filters (11 and 5 in the first and second convolutional layer, respectively) with multiple 3×3 kernel-sized filters one after another.

The input to the first convolutional layer is of fixed size 224 x 224 RGB image. The image is passed through five groups of convolutional layers. after each group, there are a Max-pooling layer.

Max-pooling is performed over a 2×2 pixel window, with stride 2. Three Fully Connected layers follow convolutional layers the first two have 4096 channels each, the third performs 1000-way ILSVRC classification and originally this contains 1000 channels (one for each class).

We have edited that Layer changing the output to 7 class labels. The final layer is the soft-max layer.[4]

3.3. GoogLeNet

The GoogLeNet model proposed in [5], is significantly more complex and deep than all previous CNN architectures. This module is based on several very small convolutions in order to drastically reduce the number of parameters. That architecture consisted of a 22 layer deep CNN but reduced the number of parameters from 60 million (AlexNet) to 4 million. More importantly, it also introduces a new module called “Inception”, which concatenates filters of different sizes and dimensions into a single new filter. The idea of the inception layer is to cover a bigger area, but also keep a fine resolution for small information on the images. So the idea is to convolve in parallel different sizes from the most accurate detailing (1x1) to a bigger one (5x5). Overall, GoogLeNet has two convolution layers, two pooling layers, and nine “Inception” layers. Each “Inception” layer consists of six convolution layers and one pooling layer. GoogLeNet is the current state-of-the-art CNN architecture for the ILSVRC challenge.[6]

3.4. Our Model

We developed our CNN model, taking inspiration from the previously models, to do face expression recognition. Our model consists in 9 convolutional layer and 3 fully connected layer. The first three convolutional layers are followed by a max pooling layer then there is a series of four convolutional layer followed by another max pooling layer. After that there are the last two convolutional layers. Finally, there the three fully connected layers preceded by an average polling layer. Besides, we also added some dropout layers between layers for regularization.

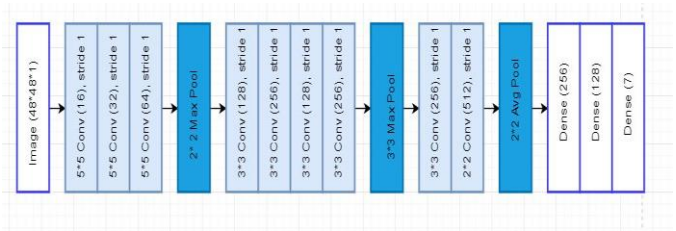

Figure 1: Diagram of the layer composition of Our Model.

4. Dataset and pre-processing

4.1. Dataset

For compare different models we made subsets starting by the csv file dataset “Kaggle challenge on Facial Expression Recognition” of 2013.[7] This dataset is composed by 35.887 (48x48 pixel grayscale) images of faces. The faces have been automatically registered so that the face is more or less centered and occupies about the same amount of space in each image. In that dataset each face is categorized in one of seven categories based on the emotion shown in the facial expression. These emotions are angry, disgust, fear, happy, sad, surprise and neutral which are respectively labeled with progressive number from 0 to 6 (images are not equal distributed for the various emotions). The provided csv contains three columns, in the first there is the numeric code that represent the emotion, the second column is a string which contains space-separated pixel values (in row major order) that compose the image, the last column indicate for which subset we will use the image: training, public_test (for validation) and private_test (for testing).

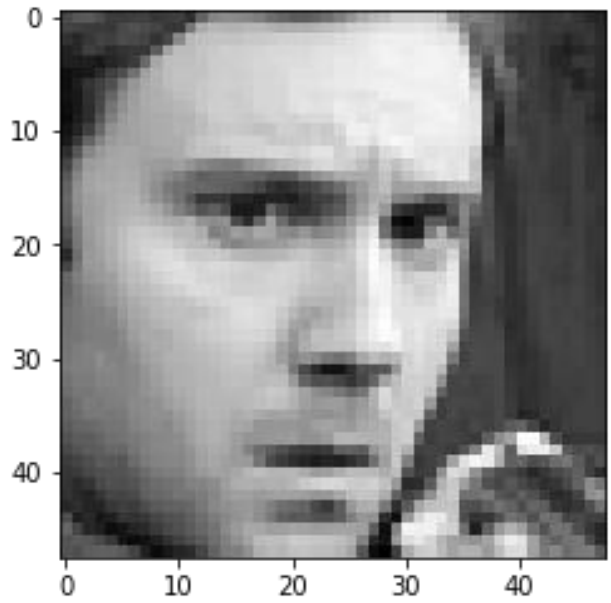

Figure 2: Image example of Kaggle Dataset.

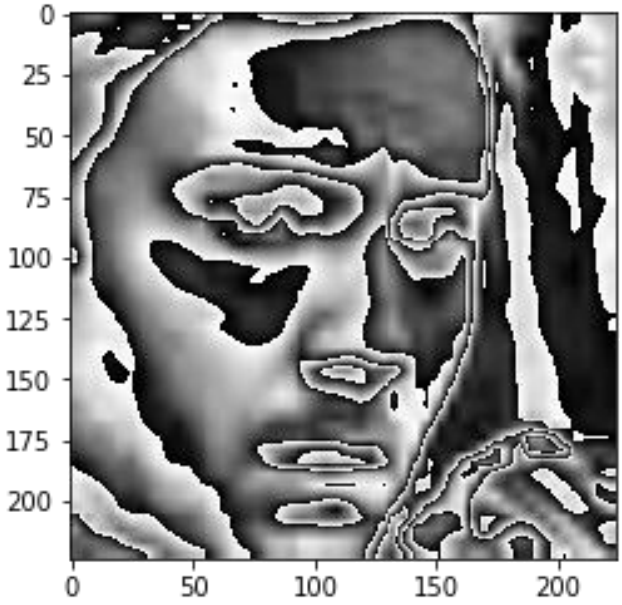

Figure 3: Preprocessed image for input to the nets.

4.2. Pre-processing

We create three different subsets according to the "Usage" column of the csv file, so we obtained 28709 images for the training dataset, and 3589 images for both the validation dataset and the testing dataset. For each subset we applied at the images a resize from 48 x 48 to 224 x 224 in order to have a compatibility with the expected input from nets followed by a central crop. Furthermore, so as to have the number of channels expected by models, we replicate the single grayscale image in two other channels. Last we normalized the images with the same mean and std of the pretrained nets with the ImageNet dataset.

5. Results

Our task in this project was to categorize each face based on the emotion shown in the facial expression in to one of the seven possible categories.
For evaluate the performances of the chosen models we take care of accuracy and loss curves and after that we study a confusion matrix on the test datasets of the models.

5.1. Hyperparameters tuning

For each model we performed a tuning hyperparameter session of 10 epochs for Alexnet and ResNet50 and 15 epochs for GoogLeNet and 30 epochs for the model that we created. In this session we choose the learning rate and batch size which give us the best accuracy. The following figures show the accuracy and the loss through the epochs on the validation set, for best parameters found previously.

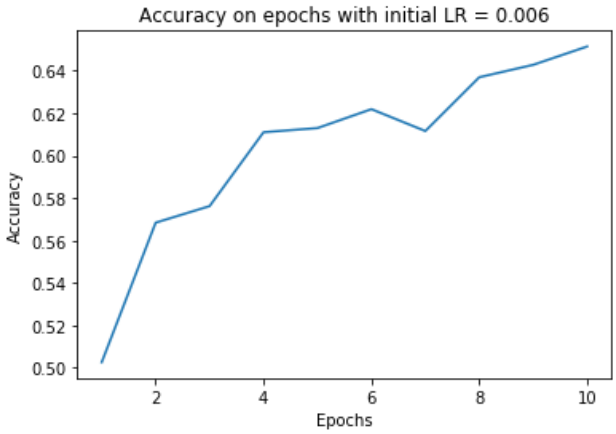

Figure 4: AlexNet Accuracy with LR 0.006 batch size = 256

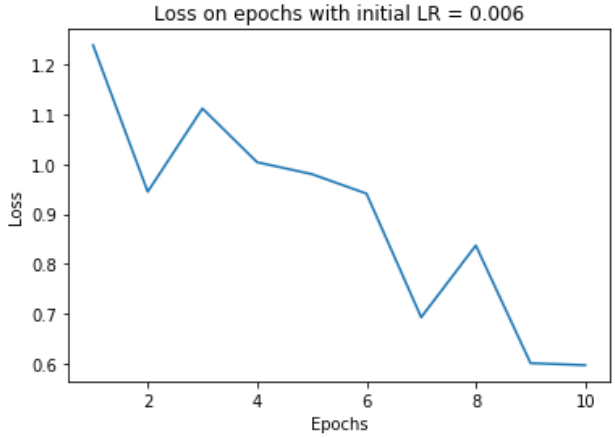

Figure 5: AlexNet Loss with LR 0.006 batch size = 256

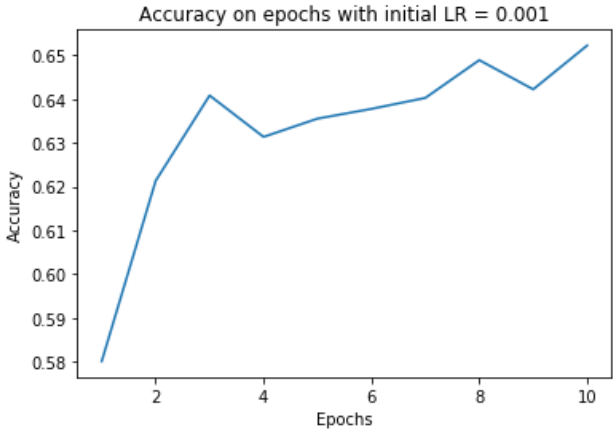

Figure 6: ResNet50 Accuracy with LR 0.001 batch size = 128

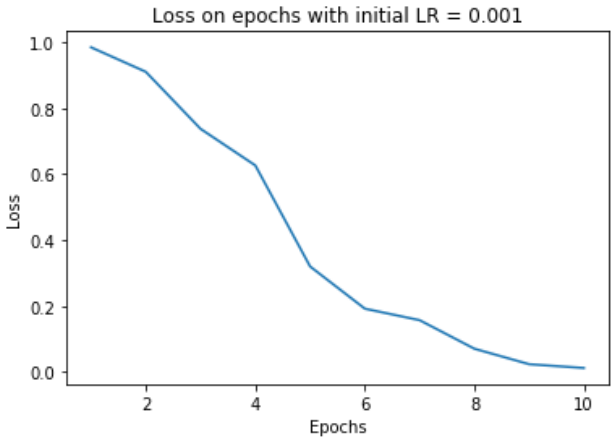

Figure 7: ResNet50 Loss with LR 0.001 batch size = 128

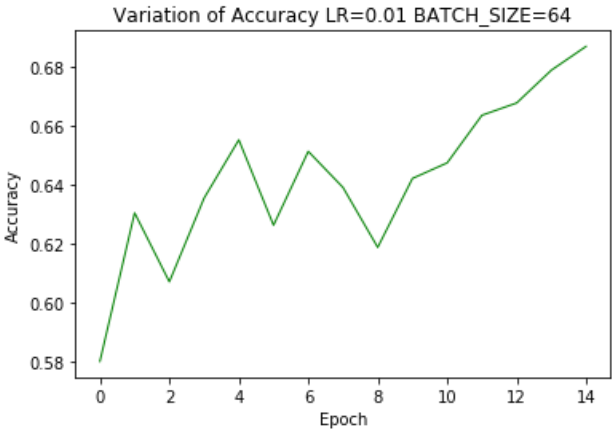

Figure 8: GoogLeNet Accuracy with LR 0.01 batch size = 64

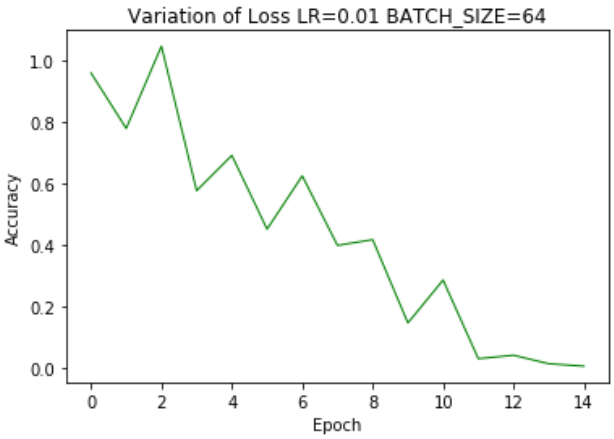

Figure 9: GoogLeNet Loss with LR 0.01 batch size = 64

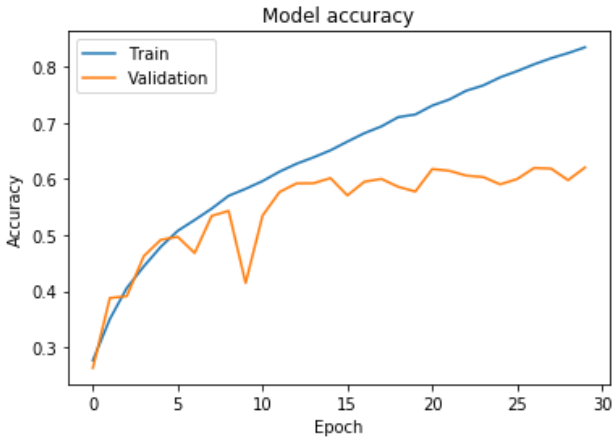

Figure 10: OurModel Accuracy with LR 0.01 batch size = 24

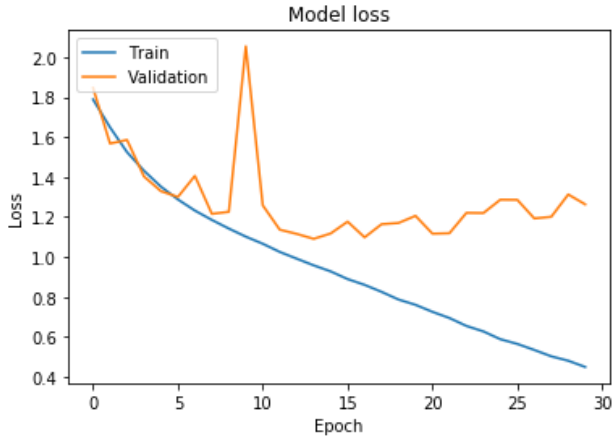

Figure 11: OurModel Loss with LR 0.01 batch size = 24

We can see that:
- for AlexNet the best accuracy is found with learning rate of 0.006 and batch size of 256
- for ResNet50 the best accuracy is found with learning rate of 0.006 and batch size of 128
- for GoogLeNet the best accuracy is found with learning rate of 0.01 and batch size of 64
- for Our model the best accuracy is found with learning rate of 0.01 and batch size of 24

5.2. Accuracy comparison

In the following table we report the accuracy found on the various subset performence (the values of train accuracy refer at the train dataset joined to the validation dataset tested after train with best values of the parameters).

	Train Accuracy	Validation Accuracy	Test Accuracy
AlexNet	78%	64%	64%
ResNet50	88%	65%	67%
GoogLeNet	96%	68%	69%
OurModel	83%	61%	62%

We can see that the accuracy on the train dataset is closed to the average (86%) for each one the model that we have analyzed. The best results that we have obtained as we expected are found for with GoogLeNet with a train accuracy of 96%.
Our model's trend is similar to AlexNet's trend.

5.3. Confusion Matrix

For another evaluation we compare the confusion matrix of the best model (GoogLeNet) with the confusion matrix of our model. The confusion matrix below represent in the rows the actual emotion and in the columns the predicted emotion. Through the diagonal line we can see where the model predict the expected emotion (true positive), in the others fields of the matrix there are the number of instance relative at one class that are classified with a wrong expected label.

GoogLeNet	Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral
Angry	288	3	50	21	69	12	48
Disgust	10	37	1	4	2	0	1
Fear	59	1	266	18	98	38	48
Happy	9	0	14	790	24	16	26
Sad	39	3	80	20	342	4	106
Surprise	8	1	32	28	3	330	14
Neutral	40	1	36	32	78	5	434

Figure 12: GoogLeNet Confusion matrix

OurModel	Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral
Angry	253	6	39	32	86	6	69
Disgust	9	36	2	3	3	1	1
Fear	64	3	203	22	119	45	72
Happy	21	1	6	728	58	16	49
Sad	57	5	43	28	314	5	142
Surprise	4	1	33	29	10	314	25
Neutral	36	2	19	31	107	12	419

Figure 13: Our Model Confusion matrix

In the previously confusion matrix, we marked the critical classes (the most common wrong predicted class) for each expected class. How we can see the behavior of the two models are similar. For instance, we can see that the class "Sad" is often miscategorized with the "Neutral" label that is the highest error that the models gives. On 594 total instances labeled "Sad" we can see that GoogLeNet give us 342 correct prediction (57,5%) and 106 wrong "Neutral" prediction (17,8%) instead our model for the same label give us 314 correct prediction (52,8%) and 142 wrong "Neutral" prediction (23,9%).

6. Conclusion and Future works

Our experiments give us pretty good results, the accuracy of OurModel on tests sets is similar to the state-of-the-art convolutional neural networks architectures. In fact, the accuracies values stay inside a range between 62% and 69%. It is our opinion that is that obtain better results is so hard. In fact, in this area of study we have not absolute truth. Get a 100% correct prediction is strictly impossible also for humans.

In the future we will like to combine that classification with a gender and age classification for get that purpose attractive on the marketing real world.

References

[1] Alexandru Savoiu and James Wong, "*Recognizing Facial Expression Using Deep Learning*", by Alexandru Savoiu et al, 2017

[2] Dhruv Amin, Patrick Chase and Kirin Sinha. "*Touch Feely: An Emotion Recognition Challange*", by Dhruv Amin, Patrick Chase and Kirin Sinha, 2017

[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, 2012 ImageNet Classification with Deep Convolutional Neural Networks ImageNet Classification with Deep Convolutional Neural Networks, by Alex Krizhevsky et al, 2012

[4] Xiangyu Zhang, Jianhua Zou, Kaiming He, Jian Sun "Accelerating Very Deep Convolutional Networks for Classification and Detection", arXiv:1505.06798 [cs.CV]

[5] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, and A. Rabinovich, "Going deeper with convolutions," in IEEE Conf. on CVPR, 2015

[6] Hoon-chang Shin, Holger Reinhard Roth et al "Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning", 2016

[7] Kaggle, Challenges in Representation Learning: Facial Expression Recognition Challenge, 2013