Human Facial Keypoints Detection with WestGate

Niccolò Vettorello
niccolo.vettorello@studenti.unipd.it

Elton Stafa
elton.stafa@studenti.unipd.it

Abstract

Facial keypoints detection is a very popular topic in Computer Science, its applications span from social networks to security cams or just think about smartphones face recognition that is used everyday by millions of people. The first objective of our project is to locate the keypoints in a given image using convolutional neural networks. The second and final goal is to apply and compare different approaches to solve the same problem, noticing if performances improve or get worse. Our proposal consists of 3 different models, all of them codenamed WestGate, the first one is a vanilla implementation of a convolutional neural network that works as our baseline. Then, we improved our baseline by adding data augmentation to the initial dataset and finally we decided to implement an Inception Model which applies convolutions in parallel. Unfortunately, we obtained the best results in term of accuracy with the convolutional neural network with data augmentation.

1. Introduction

Facial keypoint detection is a critical element in face recognition and has many real-life applications like prevent retail crime, find missing people, security for phones, medical diagnosis or tracking faces in videos. It is evident the importance to obtain a fast and precise procedure to detect facial keypoints but it comes with challenges to be solved.

- Facial features vary greatly from one individual to another, and even for a single individual, there is a large amount of variation due to 3D pose, size, position, viewing angle, and illumination conditions.
- Facial keypoints detection must be fast, if it has to be used with real time applications like smartphones security, we have to pay attention also to the time we spend both in training and testing. Also we know that deep neural networks can increase complexity and computational time and it is not what we want.

The objective of this task is to predict keypoint positions on face images. Each predicted keypoint is specified by an (x,y) real-valued pair in the space of pixel indices. The input is a set of images 96 x 96 and the output is a 30 dimensional vector representing the 15 facial keypoints. In our project, we are going to use deep structures for facial keypoints detection, which can learn well from different faces and overcome the variance between faces of different person or of different conditions to a great extent.

In fact, the baseline of our project is a convolutional neural network or CNN.

Convolutional neural network (CNN) is a class of artificial neural networks that has become dominant in various computer vision tasks, is attracting interest across a variety of domains, including radiology. CNN is designed to automatically and adaptively learn spatial hierarchies of features through backpropagation by using multiple building blocks, such as convolution layers, pooling layers, and fully connected layers.

We also tried to increase the accuracy of our project by costructing an Inception Network.

An Inception network is built by Inception Modules that are incorporated into convolutional neural networks (CNNs) as a way of reducing computational expense. As a neural net deals with a vast array of images, with wide variation in the featured image content, also known as the salient parts, they need to be designed appropriately. The most simplified version of an inception module works by performing a convolution on an input with not one, but three different sizes of filters (1x1, 3x3, 5x5). Also, max pooling is performed. Then, the resulting outputs are concatenated and sent to the next layer. By structuring the CNN to perform its convolutions on the same level, the network gets progressively wider, not deeper.

2. Related work

In this section we present different projects related to ours to demonstrate how challenging is the field of facial keypoints detection. Recent works have focused on deep architectures for this detection task since these structures can better capture the high-level features of a image, in our problem a given face. CNN have been successfully applied to various computer vision tasks such as image classification [2], object tracking [4], face verification [7], and image generation [1]. [5] proposed a deep convolutional neural network with 25 layers that compares well with state of art algorithms but requires enough compute resources and time to increase its performances. Deep convolutional networks do not focus on the time complexity but only on the correctness of the detected keypoints, so we decided to implement a CNN as our baseline and improve it with Inception modules. [6] proposed a simple solution with two models as baselines, which are realized based on simple neural network and convolutional neural network respectively and it is, in fact, the most similar to our approach.

3. Approach

In this section we give a formalization of the problem of facial keypoints detection in order to remove eventual ambiguities on the topic at hand.
Then we present the various versions of the network we built, clarifying critical points if any and giving background motivations on the ideas that led to the development of the models.

We propose the following approach for Facial Key Points Detection.

3.1. Data Augmentation

Data Augmentation is helpful when the training data is limited and increases the performance of a deep learning model by generating more training data. We have horizontally flipped the both images and their 15 keypoints. Then, we vertically stacked the new horizontally flipped data under the original train data to create the augmented train dataset. Then we applied linear contrast to all images and Gaussian blur to increase the dataset, this technique randomly blur the image using a Gaussian distribution.

3.2. Data Pre-processing

The image pixels are normalized to the range [0, 1] by dividing by 255.0. We also have filtered the dataset where there was missing values for the facial keypoints.

3.3. WestGate

To achieve better keypoints detection accuracy (lower loss), we build a convolution neural network model, codenamed WestGate, as our baseline and with the following structure:

- the input shape for our model is 96 x 96 x 1 due to image size.
- 3 convolutional layer with 3 x 3 filter, 0 padding and 2 x 2 pool.
- 3 dense layer with 128, 96, 64 neurons.
- final dense layer of size 30 which represent the output vector of facial keypoints.
- ReLu as non-linear activation function

The convolutional layer is the main part of a CNN, it performs a dot product between two matrices, the input and the filter. The filter is smaller than the input and this is intentionally because it allows the same filter to be multiplied by the input array multiple times in different points. The application of that filter systematically across the entire input image allows the filter an opportunity to discover a feature anywhere in the image.

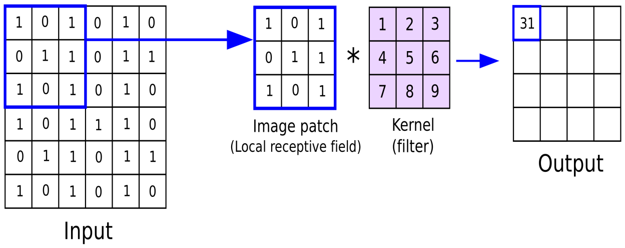

Figure 1. Example of convolution

The pooling layer replaces the output of the network at certain locations by deriving a summary statistic of the nearby outputs. This helps in reducing the spatial size of the representation, which decreases the required amount of computation and weights. There are several pooling function but we used max pooling which gather the max outputs from the neighborhood.

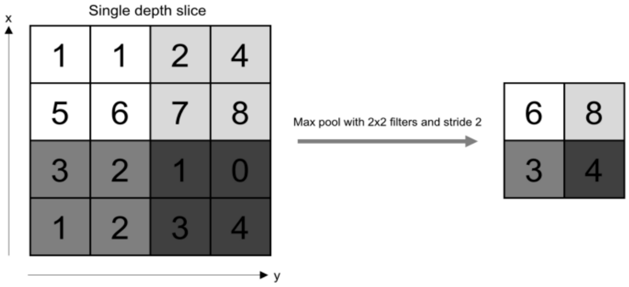

Figure 2. Max pooling

3.4. WestGate with Inception modules

As is mentioned prior in this paper, one inevitable weakness of multi-layer cnn architecture is its large amount of parameters. Due to increased layer and parameter size, the network is inclined to overfit and fails to generalize. In order to address this situation, we decided to implement an Inception network. The main part of an Inception network is the Inception layer or module. An Inception layer is a combination of all those layers (namely, 1×1 Convolutional layer, 3×3 Convolutional layer, 5×5 Convolutional layer) with their output filter concatenated into a single output vector forming the input of the next stage. This allows the internal layers to pick and choose which filter size will be relevant to learn the required information. At the end, this is the final structure of our network [3]

4. Dataset description

The dataset we use is from kaggle's facial keypoints detection competition. The dataset contains 7049 images with size of 96 × 96 pixels. Not all the images are collected with their keypoints and only 2140 out of them have positions for all 15 facial keypoints, which are used to as our dataset to train, validate and test the network.

Kaggle also provides a dataset for testing wich contains 1783 images.

The 15 facial keypoints on a given face are listed below:

left eye center	right eye center
left eye inner corner	left eye outer corner
right eye inner corner	right eye outer corner
left eyebrow inner end	left eyebrow outer end
right eyebrow inner end	right eyebrow inner end
nose tip	mouth left corner
mouth right corner	mouth center top lip
mouth center bottom lip	

Table 1. All kaggle's 15 facial keypoints

5. Experiments

In the first part of this section we discuss the various evaluation criteria used to measure performances.
Then we present the experimental results we obtained for each model.
Finally we comment the results obtained.

5.1. Evaluation criteria

To evaluate the performances of our models we decided to use well-known criteria.
In particular, for the base version of the network and for the one with the augmented data set we decided to use simple metrics to have a straightforward idea of the goodness of the network, therefore we opted for *Huber Loss* as error function and a standard accuracy estimate.
Concerning the version of the network which uses the Inception Network, we started by using the same metrics as above, but after a few runs, having gained a deeper understanding of the real complexity of the model we developed, we chose to switch to more sophisticated metrics to better understand the observed behavior: so we picked *Mean Squared Error* as error function.

5.2. Experimental results

In the following paragraph we list the results obtained by the various models.
Then we try to interpret them.

5.2.1 WestGate

The basic version of the network was trained for *100 epochs*, using a *batch size of 32*. Furthermore, we chose to use 5% of the training set as *validation set*.
These are the results:

- We obtained an average loss value of 0.5 on the training set, and an average loss value of 0.6 on the validation set;
- The accuracy value on the training set had an average of 0.8, while on the validation set it had a value of 0.7

This is an example of prediction obtained using the basic version of the network:

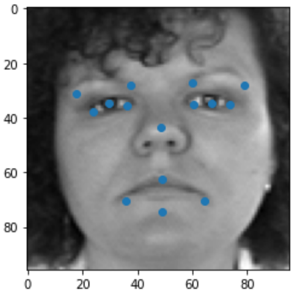

Figure 3. Example of vanilla WestGate prediction

5.2.2 WestGate with data augmentation

The version of the network with data augmentation was trained for *100 epochs*, using a *batch size of 32*. Furthermore, we chose to use 5% of the training set as *validation set*.
These are the results:

- We obtained an average loss value of 0.4 on the training set, and an average loss value of 0.6 on the validation set. During evaluation on the validation set, the loss function registered a few spikes;
- The accuracy value on the training set had an average of 0.8, while on the validation set it had a value of 0.6, registering negative peaks.

This is an example of prediction obtained using the version of the network with data augmentation:

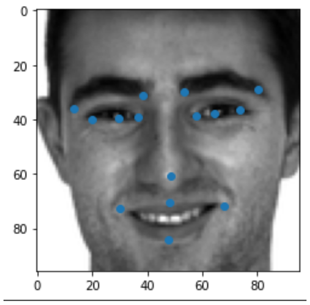

Figure 4. Example of WestGate with data augemtation prediction

5.2.3 WestGate with Inception Network

The version of the network that uses the Inception Network was trained for *300 epochs*, using a *batch size of 64*. Furthermore, we chose to use 5% of the training set as *validation set*.
We obtained an average *Mean Absolute Error* value of 2 on the training set, and an average *Mean Absolute Error* value of 4 on the validation set.
This is an example of prediction obtained using the version of the network with data augmentation:

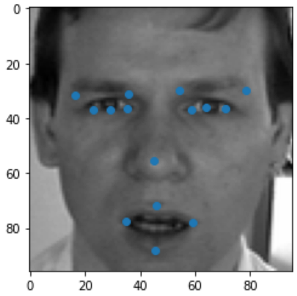

Figure 5. Example of WestGate with Inception Network prediction

5.3. Comments on results

5.3.1 WestGate

These are the representations of the results we obtained:

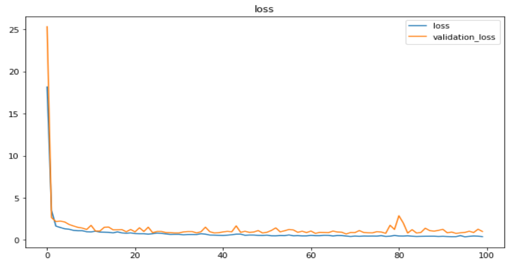

Figure 6. WestGate loss

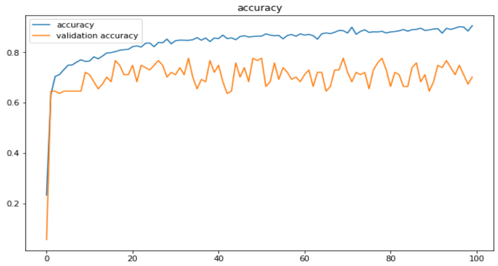

Figure 7. WestGate accuracy

Looking at the graph one can notice how the measures used behave as expected, with loss and accuracy going respectively down and up rapidly in the first few iterations, and stabilizing as epochs continue to progress.

5.3.2 WestGate with data augmentation

These are the representations of the results we obtained:

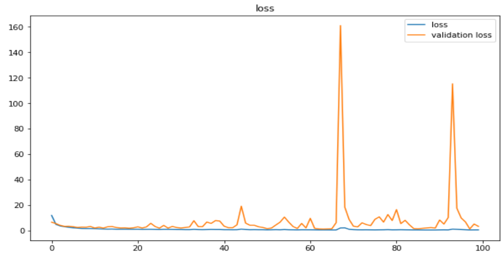

Figure 8. WestGate with data augmentation loss

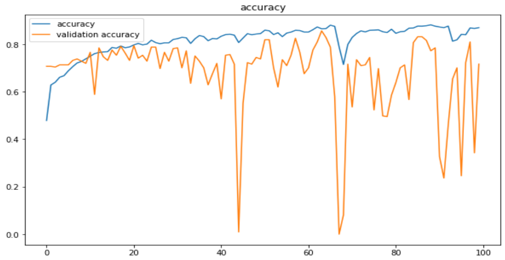

Figure 9. WestGate with data augmentation accuracy

These graphs are a bit different from the previous ones, the main difference being the presence of a few positive and negative spikes respectively in validation loss and validation accuracy.
Remembering that the training set is shuffled, these results can be explained by the fact that some batches can have by chance unlucky data for optimization.
Note however that the functions present an increasing trend (for accuracy) and a decreasing trend (for loss measure).

5.3.3 WestGate with Inception Network

These are the representations of the result we obtained:

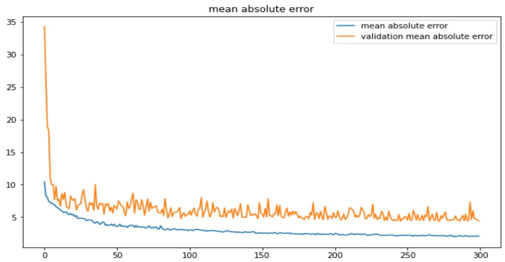

Figure 10. WestGate with Inception Network *Mean Absolute Error*

Even in this case the measurement used behaves as expected, with a rapid decrease in the first few epochs for both training set value and validation set value, and a successive stabilization around the values mentioned in the previous sub-paragraph.
Results obtained with this method do not differ substantially from the ones generated by the other two models.

6. Conclusion

The objective of this paper was to detect 15 facial keypoints given a set of images with fixed dimension 96 x 96. We proposed 3 different kind of approaches, one traditional convolutional neural network, a version of the same network but with data augmentation and an Inception network. The idea of using Inception network came with the intention to increase the accuracy of our CNN baseline, obviously this cannot be done by adding more and more layers because this will only increase the depth and the computational cost of the network. Experiments which conducted on real-world kaggle dataset have shown the effectiveness of deep structures, especially for convolutional neural network. Unfortunately, the results of the Inception network were not as expected, we found a less value of accuracy compared to the CNN. This is probably due to overfitting, caused by a small dataset for training and a greater number of hyperparameters. A future improvement would be to understand and fix this problem for the Inception network and compare the results with more and more deep learning structures.

References

[1] Soumith Chintala Alec Radford, Luke Metz. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv:1511.06434*, 2015.

[2] Geoffrey E Hinton Alex Krizhevsky, Ilya Sutskever. Imagenet classification with deep convolutional neural networks. *In Advances in neural information processing systems*, pages 1097–1105, 2012.

[3] Niccolò Vettorello Elton Stafa. Westgate inception network, 2021. https://github.com/niccolovettorello1997/VCS.

[4] Ross Girshick. Fast r-cnn. *In Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.

[5] Ranjana Vyas Naimish Agarwal, Artus Krohn-Grimberghe. Facial key points detection using deep convolutional neural network - naimishnetr, 2017.

[6] Chenyue Meng Shutong Zhang. Facial keypoints detection using neural network, 2016.

[7] Marc'Aurelio Ranzato Yaniv Taigman, Ming Yang and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, 2014.