Facial Keypoints Detection: Modern Approaches Based on ResNets

Alvise Dei Rossi
alvise.deirossi@studenti.unipd.it

Lorenzo Corrado
lorenzo.corrado@studenti.unipd.it

1. Introduction

Facial keypoints detection represents an important challenge in computer vision: in general, given a picture, the task is to correctly predict the coordinates (x, y) of a person's main facial features. The correct prediction of facial keypoints has many useful applications: for example, it plays a fundamental role in security systems based on biometrics or specialized in tracking, it can be used in the medical field, in the multimedia field (e.g. face morphing) and it's strictly linked to other important computer vision tasks like emotion detection [1].

This problem is very challenging for a large number of reasons. First of all, as we know, facial features are widely variable among different people. Futhermore, a good recognition system must be robust to all the variations in which an image can be collected such as changes of expressions, poses, illumination and backgrounds, the presence of different objects in the image, possibly occluding part of the face, just to mention a few. Finally, to capture the main details of a person's face it is also necessary to use images with sufficient resolution and this, especially using large datasets, turns out to be computationally and memory intensive.

In this context, our goal is to correctly identify 68 facial keypoints on each image we give in input to our model. A general example of the representation and position of the 68 facial keypoints can be seen in Fig.1. In order to do so, in this project we will use a particular modern type of deep CNN architectures, called ResNets. These models are able to extract high-level features and allow us to frame our problem as a regression task, where the last layer of our model has to directly predict the (x, y) coordinates for the keypoints, as opposed to many other methods using sliding window approaches. We will both evaluate the performance of these architectures as they are, trying to obtain results in a single step, and by using two further "Divide-et-impera" approaches, splitting the problem in multiple subproblems. As we will see, significant improvements can be obtained using different models specialized in the prediction of smaller portions of the face, from which a general prediction can then be derived.

The framework in which we're going to test these models will be the "300 Faces In-The-Wild Challenge" [2], organized in 2013 and 2015. We'll compare the performance of our models to those obtained from the methods proposed by the winners of the challenges [3].

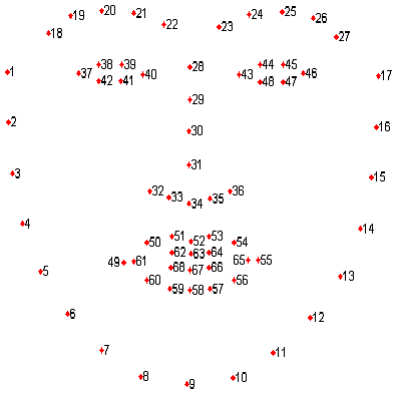

Figure 1. The 68 points mark-up used.

2. Related Works

In this section we present some examples of works that have addressed the theme of this project. In particular, we will briefly describe the models used by the 4 winners of the 2013 and 2015 challenges [3].

- Yan et al. [4] employed a cascade regression framework, where a series of regressors are utilized to progressively refine the shape initialized by the face detector. To handle inaccurate initializations from the face detector, they generate multiple hypotheses and learn to rank or combine them in order to get the final results. Parameters for both ranking and combining are estimated using a Support Vector Machine framework.
- Zhou et al. [5] proposed a four-level Convolutional Network Cascade, where each level is trained to locally refine the outputs of the previous network levels. Each level predicts an explicit geometric constraint (face region and component position) to rectify the inputs of the next levels, which improves the accuracy and robustness of the whole network structure.
- Deng et al. [6] use a multi-view, multi-scale and multi-component cascade shape regression model. Their

model learns view-specific cascaded shape regressors using multiscale HOG features as shape-index features and is optimized with a multi-scale strategy that eliminates the risk of getting stuck on local minima.

- Fan et al. [7] approach the task by proposing a deep learning system that consists of a cascade of multiple Convolutional Neural Networks (CNNs), optimized in a coarse-to-fine strategy to improve accuracy.

The inspiration for the part of our work related to breaking the main problem into multiple sub-problems is derived mainly from Zhou et al. and Fan et al. Differently from their approach, in our work deeper modern architectures and modern guidelines and tools (e.g. usage of relu, batch normalization, adam) were employed, data augmentation procedures completely replaced direct image manipulations between levels and different strategies were tested to obtain the bounding boxes for the testing set, for example by direct manipulation of the results of the single-step model.

3. Dataset and Preprocessing

As reported in Sagonas et al. [2], the dataset used comes from the aggregation of 5 different datasets that have been collected during the years:

- **LFPW:** The "Labeled Face Parts in-the-wild" dataset contains 1,287 RGB images downloaded from flickr.com, google.com and yahoo.com.
- **HELEN:** The "Helen" dataset contains 2,330 RGB images downloaded from flick.com.
- **AFW:** The "Annotated Faces in-the-wild" dataset contains 205 RGB images.
- **AFLW:** The "Annotated Facial Landmarks in-the-wild" dataset contains 25,000 RGB images downloaded from flickr.com.
- **300-W:** The "300 Faces in-the-wild" dataset contains 600 RGB images, downloaded from google.com. The images are collected in different contexts: specifically, 300 images are taken indoor and the remaining ones in an outdoor environment. The ground-truth consists of 68 landmark points for each image.

The first 4 datasets existed previously to the challenge and originally differed in the number of keypoints provided but, in order to obtain images appropriately annotated with the 68 landmark points, the authors of the challenge used semi-supervised learning techniques to re-annotate them; for further specifics see Sagonas et al. [8]. Each dataset is hence provided with the ground-truth coordinates of the facial keypoints, which will be used to train and validate our models. The aggregated dataset turns out to be quite challenging as it collects images containing different subjects, different poses and expressions, different types of lighting, background, image quality and in some cases occlusion.

An extensive preprocessing phase was needed in order to prepare the dataset. Indeed, the images within it did not only contain the faces of the subjects but often the whole body, or even multiple people. Since the goal of this work is to identify only the facial keypoints, it was necessary to crop the images. The coordinates to perform the cut were provided together with the dataset. The resulting images consisted in only the face of the subject, already centered. After the cutting phase, keypoints had to be moved accordingly. The cropped images in the dataset were also of different sizes. Therefore, to correctly implement our models, it has been necessary to resize all the images to a predetermined size. We have chosen this to be 224x224 pixels, as it is the same dimension as the preprocessed images of ImageNet in most applications, allowing the possibility to employ transfer learning. Again, to maintain the correct positioning of the keypoints, their coordinates have been rescaled proportionally after the resizing. Finally, all image pixel values have been normalized by dividing them by 255. After this preprocessing phase, the dataset was split up as follows: 3148 images were used for training, 688 for validation and 600 for testing (relative to the 300-W dataset). In Fig.2 it is possible to see an example of some images after the preprocessing phase with their associated keypoints.

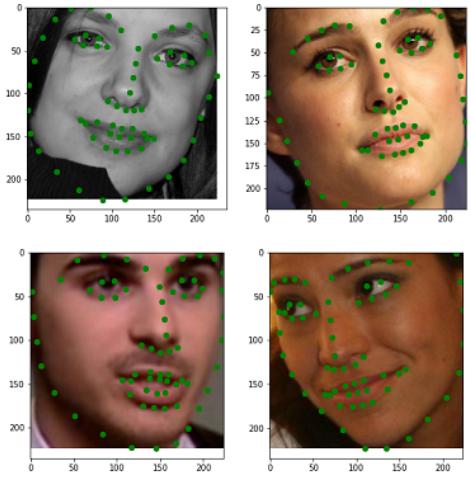

Figure 2. Examples of images after preprocessing.

4. Methods

The approaches used in this work are based on ResNet architectures, developed by He et al. [9]. Residual Networks are particular CNN architectures which use an extremely deep network, with several dozen layers. The key to being able to train such a deep network is to use partic-

ular building blocks, called residual units, that make use of skip connections (or shortcut connections). In the skip connection the signal feeding into a layer is also added to the output of a layer located a bit higher in the stack, mimicking what is commonly done also in several RNN architectures and alleviating the vanishing gradient problem that afflicts deep networks. An example of a residual unit is shown in Fig.4. A deep residual network can be seen as a stack of residual units, where each residual unit is a small neural network with skip connections.

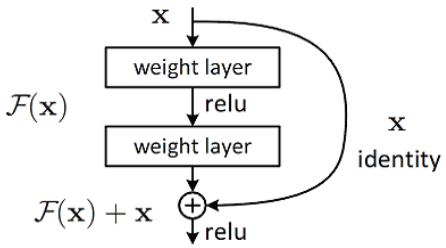

Figure 4. Residual learning: a building block.

Our **first approach** consisted in directly employing a ResNet to predict the coordinates of all the keypoints, considering the task as a simple regression problem. This approach is more straightforward than trying to incorporate human's prior knowledge into a model, taking into account local features we might expect, which tends to be challenging when much variability occurs as in this case. In a regression-based method instead, the success of the approach depends only by the regressor's ability to learn complex relations between pixel values and the appearance of facial features. The main challenge in this context is taking care of optimal training of the model, and possibly to frame the problem in simpler terms, for example by splitting it in simpler ones.

The **second approach** we used was based on a "Divide-et-impera" approach, vaguely inspired by ensemble learning. Instead of using a single model to predict all the keypoints, it's maybe possible to achieve better performance using different models specialized in identifying facial keypoints in smaller portions of the face. In particular, using the given coordinates for the ground-truth (GT) keypoints we divided and cut each face in sections, respectively for jaw, nose, mouth, right and left eye; thus, we got 5 portions for each starting image. It is important to note that in the cutting phase we added an offset with a small random noise for two reasons: the first reason is to make sure the image was correctly centered; the second and more important reason is that we wanted to avoid that the model would learn that the coordinates of some keypoints were always on the borders of the image, hence possibly not actually learning to predict the keypoints based on the shape of the face, but rather just always outputting the same coordinate. Similarly to what was previously done, once the 5 sections had been cut, it was necessary to resize them (so that all images of a section had the same dimensionality) and proportionally rescale the keypoints. After obtaining the 5 portions of the face for each image, we trained 5 different region-specific ResNets (similar to the one of the first approach). Once the predictions for every region are obtained, the image is re-

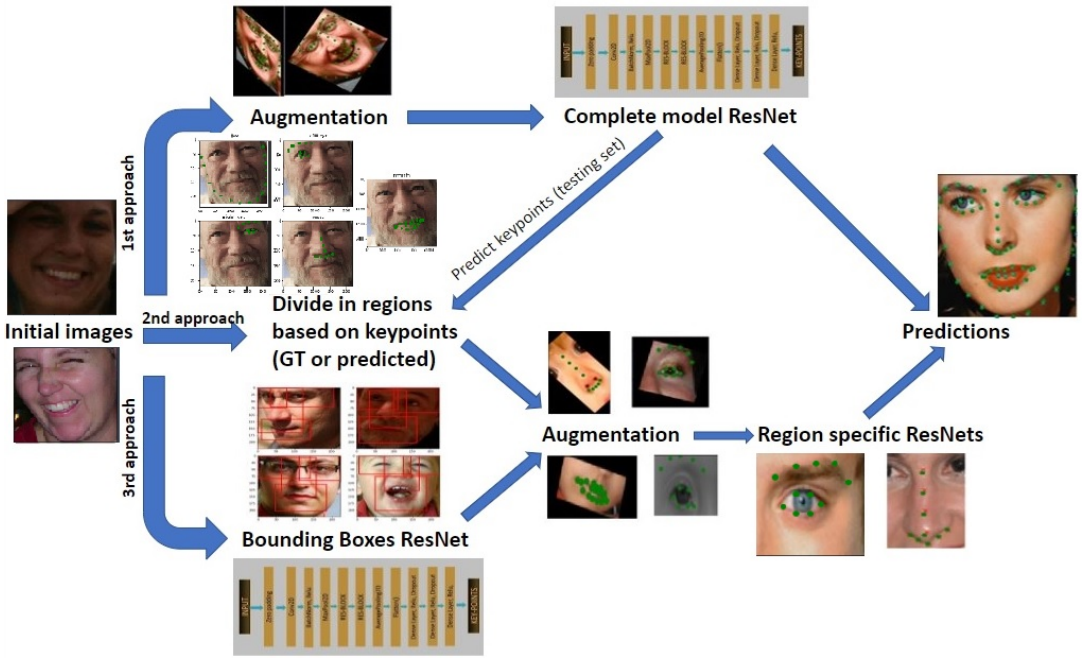

Figure 3. Summary scheme of the three methods employed.

constructed, properly moving the predictions based on the previous cuts, obtaining the final prediction for all 68 keypoints. Note that during the testing phase, since we have to imagine to not have access to the target keypoints in a real-world scenario, instead of using the target keypoints to figure out where to cut the images in 5 portions, we used the predictions of the first approach.

The **third approach** we used generalizes the concepts seen in the second. In practice, given the validity of the approach based on the prediction of keypoints on regions of the face, we tried to think of a methodology that is more automatic and general than the identification of the bounding-boxes based on the keypoints given (or predicted by the complete model in case of the test set). For this reason, we decided to use a neural network, with an architecture similar to the one used in the first approach, whose only task is to predict, for each facial region, the 4 coordinates of the rectangle that identifies the bounding-boxes (i.e. right and left eye, nose and mouth), instead of trying to predict the 68 landmarks at once. A small random offset was added in this case aswell. Once the bounding-boxes are obtained, the same region-specific models defined in the second method are used to obtain the final predictions. A summary of the approaches described is reported in Fig.3, note that some minor details are not reported.

In order to have a fair comparison with the results of the challenge, to evaluate our approaches we used the same methodology proposed in Sagonas et al. [2]. The accuracy measure for any image was obtained by calculating RMSE of the error in the point-to-point distance between the coordinates predicted by the model and the ground-truth coordinates, normalized by the interocular distance. In particular, by identifying the measured and ground-truth coordinates with $[x_1^f, y_1^f, \dots x_N^f, y_N^f]$ and $[x_1^g, y_1^g, \dots x_N^g, y_N^g]$ respectively, the error was calculated by:

$$\frac{\sum_{i=1}^{N} \sqrt{(x_i^f - x_i^g)^2 + (y_i^f - y_i^g)^2}}{d_{outer}} \tag{1}$$

Results and implementation details will be discussed by graphing cumulative error rates, as it was also done in the challenges.

5. Experiments

5.1. Setup

The setup we used in this experiment was provided by Google Cloud Platform. The code was run on a virtual machine that provided 16 vCPUs, RAM 64Gb and a Nvidia Tesla P100 GPU. This allowed us to run the code without many problems in memory allocation and to train our algorithms in an acceptable amount of time, considering the computational complexity of this task.

5.2. Implementation

We describe the architecture used in the first approach. We'll quickly mention which tests were carried out in order to define the hyperparameters, based on validation loss.

- **Input layer and first convolutional layer:** the input layer receives images of size 224x224x3, which is the final size of the images after the preprocessing phase, 3x3 padding is then added. In the first convolutional layer 64 7x7 filters were used with a 2x2 stride, followed by a batch normalization layer, relu and a 3x3 max pooling layer. The choice of the first part of this network has been due to the fact that it was observed that using pre-trained weights from the ResNet50 model trained on ImageNet sped up considerably the training phase and slightly improved the performance of the models. However we observed that initializing in the same way the weights of deeper layers didn't improve the performance, just restricting us with regards to the hyperparameters of those layers. Hence only the first part of the network is exactly equal to the standard implementation of a ResNet in Tensorflow.

- **Residual blocks:** to have the desired flexibility of choice for the number of residual units and the hyperparameters within them (number of filters, usage of Batch Normalization, padding, stride etc), we decided to code from scratch a function to add residual units to the model, at pleasure. Hence, we tested a different number of residual blocks, between 1 and 3 blocks and a different number of filters in each residual block. At the end of the testing phase, we decided to use 3 residual blocks, progressively reducing dimensionality and increasing the numbers of filters applied, from 64 to 1024. At the end of the last residual block we finally added an average pooling layer of size 2x2.

- **Dropout layer:** given the complexity of the model and the large number of parameters we added a dropout layer in order to mitigate any overfitting problems. Multiple drop probabilities were tested, the better results in terms of validation loss were obtained for a 0.5 probability value.

- **Dense layer:** in the final dense layer we used a number of neurons equal to 136 to predict the coordinates (x_i, y_i) of the 68 keypoints, with "relu" activation function. We tested also a different number of dense layers, however we noticed that adding complexity to the model in this case only worsened the performance (Fig. 5a), so we used a single final dense layer.

The optimization algorithm selected is Adam, but other algorithms such as SGD and RMSProp were tested (Fig. 5b).

The optimal initial learning rate turned out to be 0.001. We selected a batch-size equal to 16, after making a comparison between values in the range between 16 and 256 (Fig. 5c). The loss function to be optimized was the MSE and the model has been trained for 50 epochs, with early stopping, saving only the weights relative to the epoch with smallest validation loss. Finally, to improve the performance of the network, we decided to augment the starting dataset. The aim was to try to generate observations that would allow a better generalization, to address the vast variability of conditions of the images already present in the dataset. New examples were generated by varying the brightness of the images or applying shear, flips and rotations to the images. At the end of this procedure, it is possible to appreciate an increase in the performance of a model trained with the augmented training set against a model trained on the standard training set (Fig. 5d); in particular some instances which previously were badly predicted, improved substantially. This led us to use this procedure also in all other approaches explored. It is important to highlight the fact that data augmentation was rather difficult and time consuming to set up, compared to usual tasks such as image classification. In fact, whenever an image was distorted, it was necessary to rescale or move the keypoints very carefully.

For the second approach, as mentioned earlier, we built a series of models specialized in predicting smaller portions of the face. The architectures we used are almost identical to that of the previous point, in fact we only removed the last residual block and reduced the input size depending on the region we were considering. Data augmentation was extensively used in this case too. An interesting point noted during the training of the model relative to the jaw region is that the prediction of its keypoints resulted to be more imprecise (in any configuration of the model tested), compared to the keypoints predicted by the complete model. This may be due to the fact that the jaw has less defined details than other parts of the face, easier to distinguish. Possibly, compared to the other sections, having information about the relative position of the other keypoints improves the chances of the model to accurately predict the position of these points on the jaw. Therefore the keypoints of the jaw were only predicted from the complete model by all approaches. This issue was brought up in related works too. It's the reason why in the previous editions of the challenge the results were split for the case where all 68 keypoints were considered and where only the 51 keypoints non-related to the jaw were considered.

Finally in the last approach we defined another ResNet, exactly equal to the first one except for the last layer where only 16 output units are needed to define the 4 bounding boxes relative to the nose, lips, right and left eyes. Note that no bounding box was predicted for the jaw.

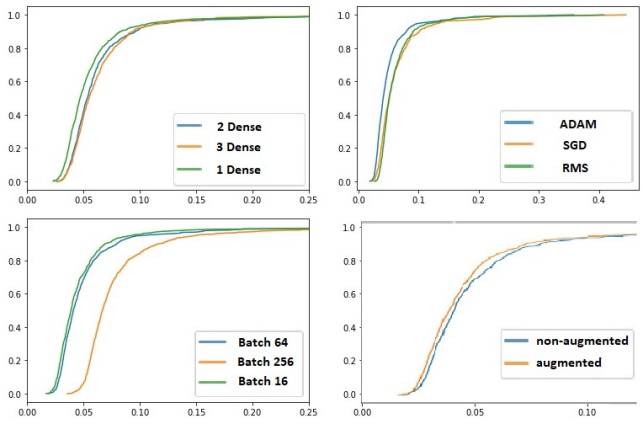

Figure 5. Cumulative error rates on validation set.
a) choice of number of dense layers; b) choice of optimizer
c) choice of batch size; d) improvement with augmentation

5.3. Results

The three approaches were finally tested against the test set. Both region specific approaches outperformed the direct approach. Defining the bounding boxes with a specific resnet (BB), rather then obtaining them with the predictions of the direct model, turned out to be slightly more beneficial. It was noted that the performance of the region specific models is highly dependent on the appropriate definition of the bounding boxes for the regions; a test was carried out to see what outcome could be obtained by defining the bounding boxes using the ground-truth (GT) coordinates of the keypoints, observing that major improvements could be obtained if more accurate methods were employed (Fig. 6).

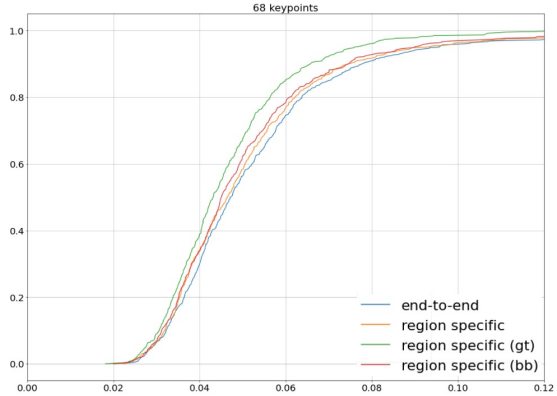

Figure 6. Comparison of the results obtained for the three approaches employed (68 keypoints).

The results are then compared to those obtained by the winners of the two editions of the challenge (Fig. 7), both in the case of 51 and 68 keypoints. Our approaches perform in a comparable way for images with low error, those at the beginning of the cumulative curve. Region specific models are actually already slightly outperforming most other

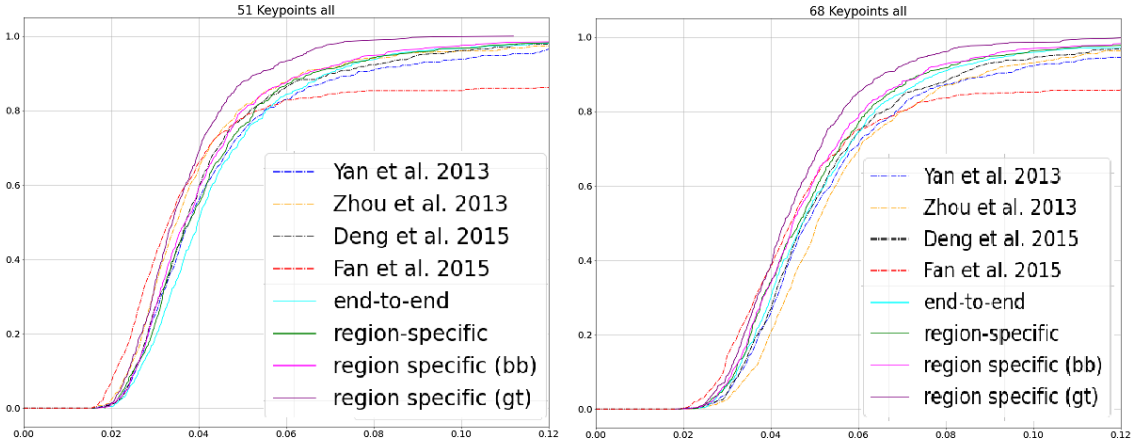

Figure 7. Results comparisons to previous works (51 keypoints left, 68 keypoints right).

methods when considering all 68 keypoints. Furthermore these methods seem to really improve over past works when we consider the errors for the more challenging images, towards the end of the curve, suggesting they're more robust than the others. This seems to be confirmed by the fact that our models perform equally for pictures taken indoor compared to those taken outdoor, which were considered by past participants to be more challenging. It should be noted, however, that there's certainly room for improvement in the implementation of some steps of the process as can be seen from the gap to the ideal error (GT) using these models.

Finally we conclude by showing some examples of the best and worst predictions we obtained (Fig. 8). As it was easy to anticipate, images with low error are well-centered, frontal with slight or no rotation of the head, bright, with no cluttering. Images with high error, on the other side, are pictures of people screaming, crying, making faces, occluded, highly rotated, too dark or too bright, with low resolution. In most of these cases a specific region of the face is responsible for most of the error committed by the overall model, corresponding to the most distorted/occluded feature.

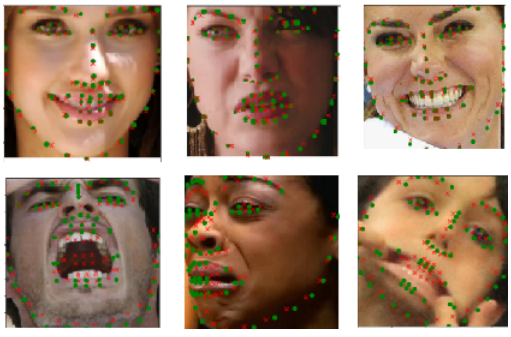

Figure 8. Best (1st row) and worst (2nd row) predictions examples. green points are the target keypoints, red crosses the predictions.

6. Conclusions and Future Works

The aim of this project was to tackle the task of facial keypoints detection. Three approaches were proposed to solve the problem making use of Resnets and data augmentation. Satisfactory results were obtained, compared to those of past participants in the challenge, especially improving the robustness. As mentioned, it could be possible in the future to improve the region specific models by defining more precisely the bounding boxes, as demonstrated when taking into consideration ground-truth keypoints to decide where to cut the images. To do so, a different approach that might be tried is to employ an object detector system (like a R-CNN based method [10] or YOLO [11]), employing sliding window approaches, trained on faces to properly detect facial features. In a real-world scenario these models could also be employed previously as face detectors [12] aswell; in fact keep in mind that the models we proposed take for guaranteed to be proceeded by a proper face detector beforehand.

A completely different approach to tackle the problem of the definition of the bounding boxes of every region,only briefly considered in our tests, could be to base the predictions using pose estimation and associating every target image to the groundtruth bounding boxes of its closest neighbors in the training set. This pose estimation process might be carried out for example making use of siamese networks based themselves on ResNets [13].

Finally, by observing the worst errors made by the models we used, we noticed that a small percentage of images in the dataset were extremely complex, collected in extreme conditions. We feel that having an even more representative training set, by including more of such examples, may be beneficial in the future in order to obtain better results in this task. Alternatively generative processes [14] might be employed to boost the image augmentation process.

Github repository related to this project can be found at [15].

References

1. B. T. Nguyen, M. H. Trinh, T. V. Phan and H. D. Nguyen, An efficient real-time emotion detection using camera and facial landmarks, 2017 Seventh International Conference on Information Science and Technology (ICIST), pp. 251–255 (2017).

2. Sagonas, Christos and Tzimiropoulos, Georgios and Zafeiriou, Stefanos and Pantic, Maja. 300 faces in-the-wild challenge: The first facial landmark localization challenge. *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 397–403 (2013).

3. Sagonas, Christos and Antonakos, Epameinondas and Tzimiropoulos, Georgios and Zafeiriou, Stefanos and Pantic, Maja. 300 faces in-the-wild challenge: Database and results. *Image and vision computing*, pp. 3–18 (2016).

4. J. Yan, Z. Lei, D. Yi, S. Li, Learn to combine multiple hypotheses for accurate face alignment, Proceedings of IEEE International Conference on Computer Vision (ICCV-W), pp. 392—396 (2013).

5. E. Zhou, H. Fan, Z. Cao, Y. Jiang, Q. Yin, Extensive facial landmark localization with coarse-to-fine convolutional network cascade, Proceedings of IEEE International Conference on Computer Vision (ICCV-W), pp. 386-–391 (2013).

6. J. Deng, Q. Liu, J. Yang, D. Tao, M3 CSR: multi-view, multi-scale and multi-component cascade shape regression, Image Vis. Comput. 47, pp. 19–26 (2016).

7. H. Fan, E. Zhou, Approaching human level facial landmark localization by deep learning, Image Vis. Comput. 47, pp. 27-–35 (2016).

8. C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, A semi-automatic methodology for facial landmark annotation, In Computer Vision and Pattern Recognition Workshops (CVPRW), pp 896-–903 (2013).

9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition, CVPR (2016)

10. Ren, Shaoqing, He, Kaiming, Girshick, Ross, Sun, Jian. , Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, IEEE Transactions on Pattern Analysis and Machine Intelligence. 39 (2015).

11. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. arXiv preprint arXiv:1506.02640, 4,5, 2015.

12. H. Jiang and E. Learned-Miller, "Face Detection with the Faster R-CNN," 2017 12th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2017), pp. 650–657, 2017.

13. Gao, Fuxun, Wang, Chaoli. Head Pose Estimation with Siamese Convolutional Neural Network, IEEE 8th Data Driven Control and Learning Systems Conference (DDCLS), 580-585, 2019.

14. Creswell, Antonia and White, Tom and Dumoulin, Vincent and Arulkumaran, Kai and Sengupta, Biswa and Bharath, Anil A. Generative adversarial networks: An overview. *IEEE Signal Processing Magazine*, pp. 53–65 (2018).

15. Dei Rossi A. , Corrado L. , Facial Keypoints Detection: Modern Approaches Based on ResNets, https://github.com/SunPy-FIS/Vision