Evolutionary Data Augmentation in Deep Face Detection

João Correia
CISUC, Department of Informatics Engineering, University of Coimbra
Coimbra, Portugal
ULHT, Universidade Lusófona de Humanidades e Tecnologias
Lisboa, Portugal
jncor@dei.uc.pt

Tiago Martins
CISUC, Department of Informatics Engineering, University of Coimbra
Coimbra, Portugal
tiagofm@dei.uc.pt

Penousal Machado
CISUC, Department of Informatics Engineering, University of Coimbra
Coimbra, Portugal
machado@dei.uc.pt

ABSTRACT

We present an evolutionary approach for Data Augmentation (DA) in deep Face Detection (FD). The approach is fully automatic and creates new face instances by recombining facial parts from different faces. We explore the selection of the facial parts that construct each new face instance using two strategies: random and evolutionary. The evolutionary strategy employs a Genetic Algorithm (GA) with automatic fitness assignment based on a pre-trained face detector. The evolutionary approach is able to find new face instances that exploit the vulnerabilities of the detector. Then we add these new instances to the training dataset, retrain the detector, and analyse the improvement of the performance of the detector. The presented approach is tested using deep object detectors, trained with instances from the CelebFaces Attributes (CelebA) dataset. The experimental results show that the presented approach improves face detection performance when comparing to base models trained using standard DA techniques. Also, the approach generates new realistic faces with interesting and unexpected features.

CCS CONCEPTS

• **Computing methodologies → Neural networks**; **Genetic algorithms**; *Object detection*;

KEYWORDS

Data Augmentation, Evolutionary Computation, Face Detection, Deep Learning, Machine Learning

ACM Reference Format:
João Correia, Tiago Martins, and Penousal Machado. 2019. Evolutionary Data Augmentation in Deep Face Detection. In *Genetic and Evolutionary Computation Conference Companion (GECCO '19 Companion), July 13–17, 2019, Prague, Czech Republic.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3319619.3322053

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-1-4503-6748-6/19/07…$15.00
https://doi.org/10.1145/3319619.3322053

1 INTRODUCTION

Recent developments in Deep Neural Network (DNN) architectures have accelerated the creation of efficient deep face detectors. However, the supervised training of face detectors requires large sets of face data, which is arduous and time-consuming to gather. Thus, techniques of Data Augmentation (DA) caught the attention of the research community and in some cases became standard in the pipeline for the training of DNNs. More specifically, in the context of Face Detection (FD) most of the existing approaches apply image filters available range from simple transformations, *e. g.* rotation, scale, flip horizontally or vertically; to more complex ones, *e. g.* adding noise, filters or distortions. However, the methods listed, are generic image processing operations that do not exploit knowledge of the underlying problem domain to generate new instances [4].

In this work, we present a DA approach that dynamically and automatically creates new face instances based on the existing ones under unconstrained conditions. Furthermore, following the framework and ideas of previous work [1, 2], the generated face instances are to the training dataset with the goal of improving the performance of the detector.

2 APPROACH

The presented approach automatically generates new face instances from a given input dataset of face images by recombining the facial elementary parts (eyebrows, eyes, noses and mouths) of different faces. Each new face instance consists of a recombination, or a blend, of different faces. Figure 1 shows new face instances consisting of facial parts from different faces blended into the same face.

In each image contained in the input dataset, we detect possible faces using a pre-trained face detector and their facial landmark points using a pre-trained facial landmark predictor.

The generation of one new face instance begins with the selection of one face among the faces in the dataset to be the *base* face, *i. e.* the face whose elementary parts (eyebrows, eyes, nose and mouth) will be replaced with parts from other faces. Then, we select one different face for each part we want to replace. These faces are the *source* faces, *i. e.* the faces that will provide the parts to be blended onto the *base*. The selection of the *source* faces considers the pose of the *base* face. That is, the *source* faces should have a head pose similar to the one of the *base* face. For each face, we calculate the Delaunay triangulation of each face using points based on its landmarks. The triangulation of these points allows us to divide each face image into triangles, which tend to cover corresponding facial features between faces. The next step is to warp each *source* face triangle, *i. e.* the pixels contained in it, to the corresponding *base* face triangle using affine transformations. This results in the *source* faces that are fully align with the *base* face. Then, we use a seamless cloning algorithm to blend one region of each warped *source* face into the *base* image. The regions of the different facial parts of the warped *source* faces are delimited with masks that are calculated using specific vertexes of the Delaunay triangles.

The selection of the *base* and *source* faces can be random or guided using Evolutionary Computation (EC). We use a Genetic Algorithm (GA) to find new face instances that, potentially, enrich the training dataset and thus improve the performance of the resulting detector. The genotype consists of a tuple of integers that encode a *base* face as well as one *source* face for each facial part (left eyebrow, right eyebrow, left eye, right eye, nose, and mouth). The phenotype consists of the composite of the facial parts encoded in the genotype. The crossover operator exchanges facial parts between two faces. The mutation operator replaces genes of the faces with other selected at random.

The evolutionary process is guided with an automatic fitness function that employs a face detector to find faces that are no longer detected as such, *i. e.* exploits of the detector. Therefore, the fitness of each face instance is inversely proportional to the difference between a preset target confidence, or activation, of the detector and the confidence of the detector in the predicted bounding box that is more similar to the grown-truth bounding box. The approach performs one evolutionary run per face to be evolved. For each run, a *base* face is selected from the input dataset and the evolutionary run finds the combination of facial parts that, when blended into it, obtains the best fitness.

3 EXPERIMENTAL RESULTS

To evaluate the impact of the presented approach in a FD scenario, we conducted experiments using three setups. In *setup1*, the baseline detector is trained with a baseline dataset consisting of a subset of the CelebA dataset. In *setup2*, we apply our approach to randomly generate new instances, doubling the number of instances in the baseline dataset and then retraining the baseline detector. In *setup3*, we use a GA to evolve new instances while using the baseline detector to assign fitness. Similar to *setup2*, in *setup3* we evolve new instances until we double the number of instances in the training dataset. The detector used is the "Tiny You Only Look Once (YOLO) v2", a deep single shot detector that uses a cell-grid input and anchor system to provide fast and accurate detections in real-time [5]. In each setup, we trained 10 different detectors under the same conditions but with different random weights initialisations.

FD comprises at least two distinct tasks: classify the occurrence of a face in an image and determine its location and size. To measure the performance of face detectors, we rely on Mean Average Precision (mAP) based on the Intersection over Union (IoU) of the detections, which is a measure to assess object detection results used on several benchmark and competitions [3]. The IoU threshold used for calculating the mAP was set to 0.3. We have tested our approach in the Face Detection Data set and Benchmark (FDDB) dataset containing 5171 faces in 2845 images taken from the Faces in the Wild dataset [3].

The results in terms of mAP indicate: **0.04** for *setup1*; **0.24** for *setup2*; and **0.34** for *setup3*. *setup3* has the best average result, with

Figure 1: Faces generated with the presented approach using parts of different faces blended into the same face. More results at `cdv.dei.uc.pt/evo-data-augmentation/`.

an increase of mAP over the *setup1* of 0.3 on average and 0.1 over the *setup2*. Overall, the results suggest that *setup2* and *setup3* have impact on the dataset that was defined to train models used in *setup1*. The results indicate that: *i)* adding new random instances (*setup2*) to the training dataset improves the detector performance; *ii)* using EC to evolve instances (*setup3*) results in instances that exploit the vulnerabilities of the DNN detector; and *iii)* adding evolved instances to the training dataset improves the detector performance, significantly more than when using random instances.

More details about the experimental results can be found at `cdv.dei.uc.pt/evo-data-augmentation/`.

4 CONCLUSIONS

We presented an approach for the automatic DA in FD. We tested the approach using the YOLO deep detector trained with instances from the CelebA dataset. The experimental results show that the approach improves FD performance when comparing to base models trained using typical DA techniques. Also, looking at the faces generated, the approach is able to generate new realistic faces with interesting and unexpected features.

5 ACKNOWLEDGEMENTS

We would like to express our gratitude to NVIDIA for providing us with one GPU to support our research.

REFERENCES

[1] João Correia, Penousal Machado, and Juan Romero. 2012. Improving haar cascade classifiers through the synthesis of new training examples. In *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference companion - GECCO Companion '12*. ACM Press, New York, New York, USA, 1479. https://doi.org/10.1145/2330784.2331001

[2] João Correia, Tiago Martins, Pedro Martins, and Penousal Machado. 2016. X-Faces: The eXploit Is Out There. In *Proceedings of the Seventh International Conference on Computational Creativity (ICCC 2016)*, François Pachet, Amilcar Cardoso, Vincent Corruble, and Fiammetta Ghedini (Eds.). Sony CSL Paris, France, 164–182.

[3] Vidit Jain and Erik Learned-Miller. 2010. *FDDB: A Benchmark for Face Detection in Unconstrained Settings.* Technical Report UM-CS-2010-009. University of Massachusetts, Amherst.

[4] Iacopo Masi, Anh Tuážěn Trážğn, Tal Hassner, Jatuporn Toy Leksut, and Gérard Medioni. 2016. Do We Really Need to Collect Millions of Faces for Effective Face Recognition?. In *European Conference on Computer Vision (ECCV)*. 579–596. https://doi.org/10.1007/978-3-319-46454-1_35

[5] Joseph Redmon and Ali Farhadi. 2016. YOLO9000: Better, Faster, Stronger. *CoRR* abs/1612.08242 (2016).