A Data Augmentation Methodology to Improve Age Estimation using Convolutional Neural Networks

Ítalo de Pontes Oliveira, João Lucas Peixoto Medeiros, Vinícius Fernandes de Sousa
Adalberto Gomes Teixeira Júnior, Eanes Torres Pereira, Herman Martins Gomes
Federal University of Campina Grande (UFCG)
Email: italooliveira@copin.ufcg.edu.br, joao.medeiros@ee.ufcg.edu.br, viniciusfernandes@copin.ufcg.edu.br,
adalbertojunior8@gmail.com, eanes@dsc.ufcg.edu.br, hmg@dsc.ufcg.edu.br

***Abstract*—Recent advances in deep learning methodologies are enabling the construction of more accurate classifiers. However, existing labeled face datasets are limited in size, which prevents CNN models from reaching their full generalization capabilities. A variety of techniques to generate new training samples based on data augmentation have been proposed, but the great majority is limited to very simple transformations. The approach proposed in this paper takes into account intrinsic information about human faces in order to generate an augmented dataset that is used to train a CNN, by creating photo-realistic smooth face variations based on Active Appearance Models optimized for human faces. An experimental evaluation taking CNN models trained with original and augmented versions of the MORPH face dataset allowed an increase of 10% in the F-Score and yielded Receiver Operating Characteristic curves that outperformed state-of-the-art work in the literature.**

***Keywords*-Data Augmentation; Age Estimation; Deep Learning; Fiducial Points; Face Detection**

I. Introduction

In recent years, there has been great advances in image recognition and classification due to the progress on training large neural network models. Not surprisingly, deep learning approaches are among the solutions with best results found in the literature [1]. Several factors have contributed to that, being the increase in processing power of General Purpose Graphics Processing Units (GP-GPUs) an important one, which allows faster training of neural network models with large number of stacked layers (e.g., Deep Convolutional Neural Networks - CNN) and large training sets.

Models with many free parameters (such as CNN) require a large volume of data for training. More specifically, when such models are applied to the face analysis area, there is a crescent need for large face image datasets that are not currently available. Various approaches have been proposed for general data augmentation with the purpose of generating better classification models, including CNN [2]. However, they only consider very simple transformations on the input images, such as changing the view point, flipping, cropping or making color changes (e.g. [3]), and do not take advantage of specific knowledge regarding the problem at hand.

Within the above context, this paper proposes a novel methodology based on Active Appearance Models in order to perform data augmentation on human face datasets, with a special focus on training CNNs classifiers for face image age estimation. Formally, the data augmentation term refers to methods for constructing iterative optimization or sampling algorithms via the introduction of unobserved data or latent variables [4], which are often utilized to generate additional training data without introducing extra labeling costs.

In this paper, we present a novel approach to perform data augmentation which was validated using a CNN model trained with original (about 50K images) and augmented ($>$ 1M images) versions of the MORPH face dataset. Data augmentation allowed an increase of 10% in the F-score and yielded Receiver Operating Characteristic curves that outperformed state-of-the-art work in the literature of age estimation from human faces.

This article is organized as follows: Section II describes the related work on data augmentation techniques. The CNN operation is briefly explained in section III. CNN was used to validate the proposed methodology for data augmentation. Section IV details how the proposed approach works. An experimental evaluation, implementation details and results are discussed in Section V. Finally, Section VI presents the conclusions of the work.

II. Related Work

Dosovitskiy *et. al.* [5] investigated the use of data augmentation as the main component of an unsupervised feature learning architecture. Their approach obtains random patches of size 32×32 pixels from different images and applies random transformations combining four types of variations that include: translation, scale, color and contrast variation. A limitation of that work was the absence of a comparison of the accuracy scores with and without data augmentation. Another similar work is that of Gerke *et. al* [6], in which the data augmentation technique was applied to obtain random patches from the images to increase the size of the dataset in a ratio of $1 : 10$. Classifier accuracy, in a problem of soccer jersey number recognition, has increased more than 10% when using the augmented dataset.

Data augmentation was used by Chen *et. al* [7] to reduce classifier overfitting. Translations and horizontal reflections have been applied to dataset images. On the training phase, the data augmentation process extracts random 224×224 patches from the 256×256 pixels images, including their horizontal reflections. The above discussed strategies do not consider

semantic information of the data, only changes in the point of view of objects are considered, like the linear transformations made in the work of van Dyk and Meng [4]. Zhu *et. al.* [8] also followed the same data augmentation principle, by applying random rotations, translations, scaling, noise levels, and by flipping dataset images with the purpose of increasing dataset size, thus allowing the classifier to be trained with additional views of an object.

In the detection scheme proposed by Farfade *et. al* [9], traditional techniques for data augmentation have also been employed, which consisted in generating randomly flipped and sampled sub-windows of the images. The AlexNet CNN [3] was used to train the model and obtained one of the highest scores in FDDB dataset [10].

McLaughlin *et. al.* [11] proposed a method for data augmentation that consists in manually creating a mask of people in the images and replacing the background with a different (simulated) one, in order to avoid bias in a person re-identification task using scenario information. In their experiment, the proposed technique was compared with other traditional techniques to do data augmentation like the ones described above (e.g. including crops, flips, rotations, color changes and affine transformations, and the combination between them). The technique was validated using the CNN LeNet [12] and it was observed that the proposed method was the most effective of all tested.

Contrasting to the above reviewed work, our approach for data augmentation is tailored at the face classification scenario, differently from [5], [7], [6], who applied traditional techniques to make data augmentation in different tasks, such as object recognition, glaucoma detection and digit recognition. Huerta *et. al* [13] and Fernández and Prati [14] have proposed a method based on descriptors for age estimation and present a comparative evaluation of various age estimation algorithms. We use this baseline to compare our approach to doing data augmentation and validate the method.

In the work of McLaughlin *et. al* [11] it was observed that the use of a specific approach for data augmentation is more effective than naïve transformations. Aligned with that observation, our work takes into account intrinsic information about human faces (facial landmarks or fiducial points delimiting facial components, such as mouth, nose and chin) in order to generate an augmented dataset that is suitable to train a CNN for age classification, by creating photorealistic smooth face variations based on Active Appearance Models optimized for human faces. Moreover, the method is beneficial to other face classification scenarios, where augmenting the diversity of a dataset may help reduce classifier bias.

III. Background: Convolutional Neural Network

Among existing architectures for deep learning, Convolutional Neural Networks (CNN) stand out as a relevant solution for image recognition applications. CNN allows computational models that are composed of multiple processing layers representations of data with multiple levels of abstraction.

Essentially, convolutional layers with nonlinear activation functions compose CNN. The output convolutions are computed from a previous input layer. Convolutional layers apply distinct filters and associate their results with the output. More specifically, the layer's parameters consist of a set of learnable filters, operating on small receptive fields that extend through the full size of the input. A sequence of these layers is responsible for extracting features, and turning abstract low-level features into high-level ones. The network is responsible for learning the filter weights, allowing problems with different image databases to use the same network. Nevertheless, the use of multiple convolution layers will decrease the network processing speed, due to the number of operations that each layer perform [15].

A number of deep learning frameworks implementing CNN models are publicly available. Among those frameworks, three have been identified of particular interest to the present research: Caffe [16], Torch [17], and Theano [18]. All frameworks allow training and testing models in CPU and GPU.

Table I presents a comparison of the frameworks. Caffe has been chosen because it presents a complete set of functions for training and testing deep learning models, has sufficient documentation and tutorials available, a large community of users, and pre-trained reference models (e.g., the AlexNet ImageNet model [3]), making it simple to leverage novel solutions using deep learning. Caffe is simple to use while it is computationally efficient, since its implementation is completely in C++. Furthermore, Caffe has a modular design, with many layer types, learning functions, among others, which allows developing new solutions with flexibility.

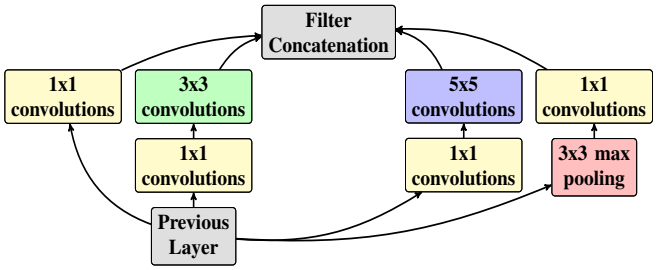
Table I
Comparison of open source popular deep learning frameworks that run in CPU and GPU.

Framework	Core language	Binding(s)	Pretrained models
Caffe [16]	C++	Python MATLAB	✓
Torch [17]	Python	-	-
Theano [18]	Lua	-	-

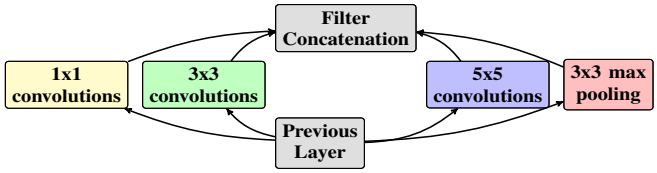
Among the CNN models implementable in the Caffe framework, the GoogLeNet [19] was selected. This model obtained the first place in the ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC2014) [20]. Besides, GoogLeNet requires less memory and processing time for training and performing predictions when compared to other CNN models. Additionally, training large datasets may be accomplished with the help of GPU programming.

The core of GoogLeNet is the inception style convolution module which is designed to relax the established principle that CNNs require large data sets and high computing power for training. The inception module allows increasing the depth and width of the network while maintaining computational costs under control. GoogLeNet's inception module is shown in Figure 1, compared to the traditional (naïve) inception version. By performing 1×1 convolutions it achieves dimension

reduction in the number of filters relative to the input, which decreases the computational load and allows implementations of larger (deeper) networks. Input layer receives training images with the size of 256×256 pixels. Deep learning network input data specification has a naïve data augmentation strategy, which extracts large random crops (e.g. 224×224 pixels) of the input image during training. For testing, only a centered crop is applied.



(a) GoogLeNet Inception module.



(b) Traditional (naïve) Inception module.

Figure 1. Models comparison based on [21].

The inception architecture has two main advantages: i) by employing filters of different sizes at each layer, it retains more accurate spatial information; moreover ii) it significantly reduces the number of free parameters of the network, making it less prone to overfitting and allowing it to be deeper than traditional architectures [22].

IV. Overview of the Proposed Methodology

This article proposes a methodology to perform data augmentation in the context of age estimation from face images. Figure 2 shows a flow diagram of the proposed methodology, which is divided into four modules. In the first step, data augmentation, fiducial points are detected in face images of a labeled dataset. This process returns the coordinates of facial features (like the eye corners and center of the pupil, the nostrils, among others). These coordinates are used in a process that systematically generates controlled facial deformations that are saved in an augmented dataset. The pre-processing module detects, crops and normalizes the faces to a standard size. This modules is applied before training and making age predictions as well. The training module is responsible for generating the CNN model based on the augmented dataset. Finally, the prediction module takes test input images after being preprocessed and uses the trained model to output a face age prediction.

A. Data Augmentation

1) Fitting Fiducial Points: Facial landmark point extraction is a fundamental step in facial image representation and analysis. Active Appearance Model (AAM), originally proposed by

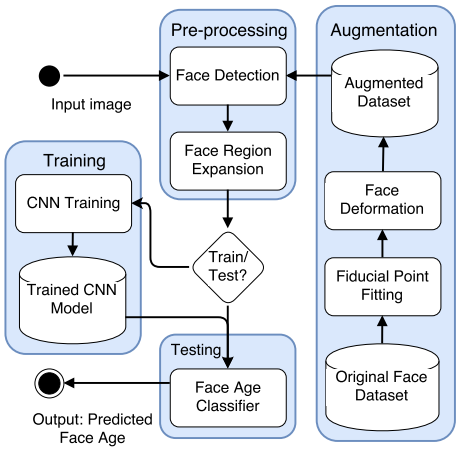


Figure 2. Flow diagram of the proposed methodology.

Cootes *et al.* [23] is a powerful object description method that is commonly used for facial landmark points extraction [24], facial action unit extraction [25], medical image segmentation and analysis [26]. The idea behind AAM is to represent a visual object (e.g. facial image) as a combined linear model of shape and texture (appearance) obtained from a set of manually labeled training images. This model is used to represent an instance of the object in a novel image. AAM descends from the Active Contour Models and Active Shape Models (ASM) [27]. Contrary to the ASM, the AAM forms a statistical model of shape and texture together. Furthermore, AAM gains a priori knowledge through an observation of the shape and texture variation across a training set.

AAM matches shape and texture simultaneously, which makes the fitting process more robust to illumination and shape variations when compared to the ASM technique [28]. AAM performs well for face images under simple backgrounds, but may present some difficulties when fitting images with cluttered backgrounds.

In this study, for face fiducial points detection, we employed the AAM library AAMLibrary [1]. This library has functions to support the two AAM stages: model learning and model fitting. The model learning phase takes as input a set of training images labeled with (in our case, facial) landmarks. The MUCT public dataset [29] (a sample of images from this database is shown in Figure 3), was used for training the AAM model. This dataset has lighting, age and ethnicity variations, consisting of 3,755 faces with 76 manual landmarks, in which each landmark may be defined as a point on the (x,y) coordinate in image, $[x_i, y_i], i = 1, 2, ..., v$, where v is the total number of landmarks. The collection of landmarks of one image is treated as one observation from the random process defined by the shape model, $S = [x_1, y_1, x_2, y_2, ..., x_v, y_v]$, in which S is the shape models and x_v, y_v are the v_{th} landmarks of the image. Eigen-analysis is applied to the observation set and the resulting linear shape model represents a shape

[1] https://github.com/greatyao/aamlibrary

according to the following equation:

$$s(P) = s_0 + \sum_{i=1}^{n} p_i s_i, \tag{1}$$

in which s_0 is the mean shape, s_i is the i-th shape basis, and $p = [p_1, p_2, ..., p_n]$ are the shape parameters. After the shape

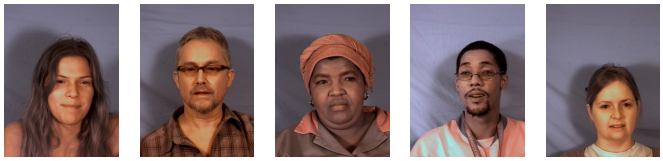


Figure 3. Sample of MUCT database face images used for AAM training.

model is produced, the next step is to produce the appearance model. For that purpose, each training facial image is warped to produce the mean facial shape s_0 and the shape-free facial appearance A, which consisted of the intensities of the warped input image modeled by a linear combination of the mean facial appearance A_0 and m facial appearance variation vectors A_i, as defined in the following equation.

$$A(P) = A_0 + \sum_{i=1}^{m} \lambda_i A_i, \tag{2}$$

in which, $\lambda = [\lambda_1, \lambda_2, ..., \lambda_m]$ represents the appearance parameter vector.

In the fitting phase, the goal is to find the shape and appearance parameter vectors that minimize the errors between the synthesized face image and the warped input face image. The objective function is defined as:

$$\sum_{x \in p(s_0)} \left[A_0(x) + \sum_{i=1}^{m} \lambda_i A_i(x) - I(W(x;p)) \right]^2 \tag{3}$$

in which W represents the warping function to change the point location from point x in the input face image coordinate to point $W(x;p)$ in the synthesized image coordinate.

Figure 4 contains an example of the final results of AAM fitting using images from MUCT database. In that figure it is possible to see the resulting facial points fitting in the eyebrows, eyes, nose, mouth and jaw.

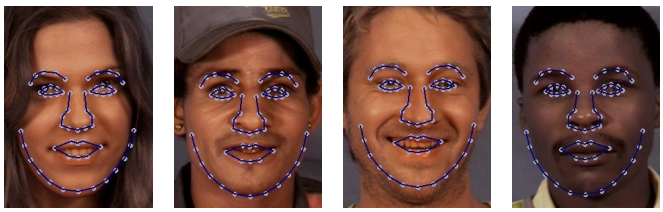


Figure 4. Fitting result using Active Appearance Model.

2) Face Deformation: In order to perform face deformation we use the resulting points of AAM fitting as deformation handles for the method proposed by [30]. The imgwarp [2] library provides an implementation of the above method. For deformation control, these handles may take the form of points, lines, or polygon grids.

Formally, image deformation can be defined as a function f that maps points in a undeformed image to points in a deformed one. Given an image with a set of labeled points p that are mapped to new positions q, for f to be useful for deformations it must satisfy the following properties [30]:

- The handles p should map directly to q under deformation. (i.e; $f(p_i) = q_i$);
- f should produce smooth deformations;
- ff the deformed handles q are the same as the p, then f should be the identity function. (i.e; $q_i = p_i \Rightarrow f(v) = v$).

Schaefer *et al.* [30] constructed a function that satisfies the above properties using the moving least squares technique on various classes of linear functions. Given the set of control point pairs, an affine function $l_v(x)$ is determined for each point v by minimizing the following expression:

$$\sum_i w_i \left| l_v(p_i) - q_i \right|^2 \tag{4}$$

in which, p is a set of control points, q is the deformed positions of the control points, p_i and q_i are row vectors and the weights w_i have the form:

$$w_i = \frac{1}{|p_i - v|^{2\alpha}}. \tag{5}$$

3) Face Deformation Variations: The proposed approach to perform data augmentation is based on the application of 2D deformations to all images of the dataset, considering a set of heuristically defined variations. The appearance of nose, chin and jaw are modified together with neighboring regions in order to create a smooth effect. Eyes and eyebrows were also considered for augmentation, but some empirical experiments we performed revealed that augmentation based on those features did not cause significant impact on the age classifier.

Figure 5 presents the location of the fiducial points considered in this work. Points from 47 to 57 correspond to nose, points from 4 to 8 correspond to the chin and the jaw is represented by points 0 to 12.

The objective of the proposed data augmentation is to create new faces from the existing ones, thus expanding the amount of data available for training and improving age prediction performance. Small deformations which do not subjectively impacted in the face age appearance have been inserted. A set of 27 variations was obtained for the dataset images by combining 3 distortions of 3 face components:

[2]https://github.com/cxcxcxcx/imgwarp-opencv

- **Nose**: large, normal and thin;
- **Jaw**: large, normal and thin;
- **Chin**: squared, normal and triangular.

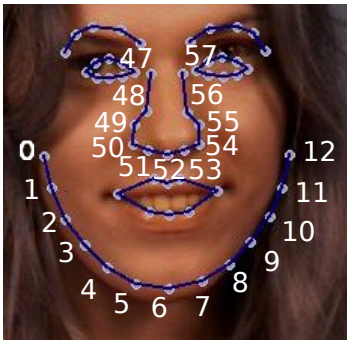
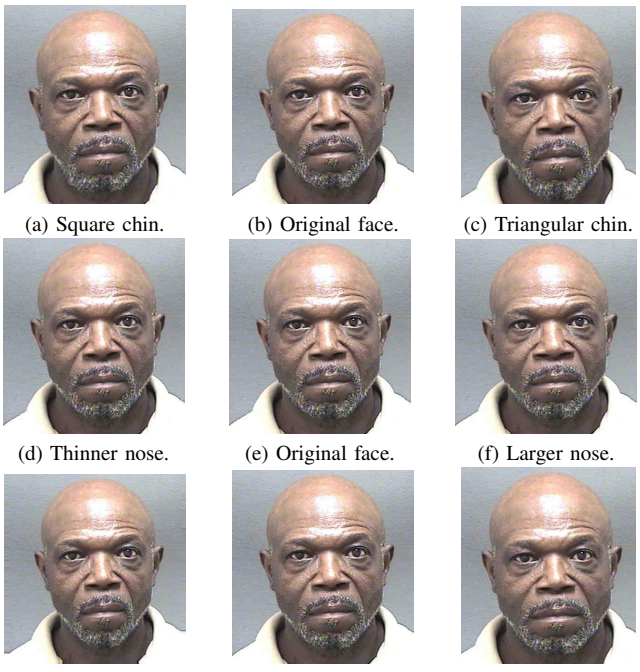


Figure 5. Enumerated points in fitting result of Active Shape Models.

An example of distortion results is illustrated in Figure 6. Subimages a-c) show chin variations; d-f) show nose variations; and g-h) show jaw variations. In subimage a) the chin is more quadrangular than the one in the original face b). In subimage c) the chin has a triangular appearance. The nose in subimage d) is larger than the one in the original face e), while the noise in f) is thinner. Similar changes occur to the jaw in subfigures g), h) and i). The aim of those changes is to increase the variability of physical facial characteristics and to decrease the difference between ethnic groups as in [31].



(a) Square chin. (b) Original face. (c) Triangular chin.

(d) Thinner nose. (e) Original face. (f) Larger nose.

(g) Thinner jaw. (h) Original face. (i) Larger jaw.

Figure 6. Examples of face variations synthetically generated.

A deformation factor and displacement rule are applied in order to define the new positions of each fiducial point found. In Equation 6 the deformation function $f(p)$ is defined.

$$f(p) = p + [dr(p) * k] \tag{6}$$

where p is the set of labeled points, k is a constant deformation factor and $dr(p)$ is the displacement rule, applied to each of the points, as described in Tables II, III and IV, for deforming points in the jaw, nose and chin, respectively. After obtaining the new fiducial point positions, the deformation function described in subsection IV-A2 is applied. A low deformation factor ($k = 3$) was employed in order to prevent distorting the facial features too much, as it can be seen in Figure 6.

In order to shrink or to expand the jaw, the jaw coordinates were translated towards the center of the face or away of that, respectively. Table II contains the offset parameters of this transformation. Both shrink and expand rules have been visually inspected on a number of image with no undesirable effects. The same principle was applied to enlarge or shrink the nose, as presented in Table III. Table IV contains the parameters needed to make the chin more elongated or more triangular. Only the vertical coordinates were modified.

B. Preprocessing

1) Face Detection: After generating the augmented dataset, it is necessary to apply a face detector to cut the face region and to use only this region to train the classifier. The algorithm used to make face detection was the Pixel Intensity Comparisons Organized (PICO) [32] which is an object detection based on the Viola and Jones object detection framework [33] - for which a publicly implementation is available in Github [3].

The face detection algorithm scans the image with a cascade of binary classifiers positioned at all coordinates and scales possible. An image region is classified as a face if it successfully passes through all cascade levels. A cascade level is formed by a binary classifier, which consists of an ensemble of decision trees with pixel intensity comparisons as binary tests in their internal nodes. The learning process consists of a greedy regression tree construction procedure and a boosting algorithm, that obtain one of the best scores for face detection on the challenging FDDB dataset [10].

2) Face Region Expansion: The crop applied by GoogLeNet CNN is 224×224 pixels and the input image received by Caffe Framework is 256×256 pixels, this is 14% bigger than crop size. So, a face region expansion has been performed in order to compensate for this difference. This process is illustrated in Figure 7, in which the top left coordinate of the face (Point A), becomes the Point B. The size of the bounding box increases from S_a to S_b while keeping the same center and enlarging the face region in 14%. When the face is detected near the limits of the image and the increased region exceeds the image boundaries, the original face is mirrored in eight directions and a face crop is applied, as it can be seen in Figure 8.

[3]https://github.com/nenadmarkus/pico

Table II
MODIFICATIONS IN FIDUCIAL POINTS OF JAW.

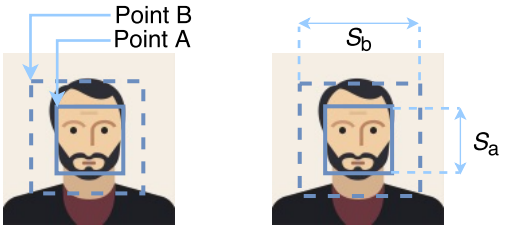
Fiducial points	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Coordinates	**(*x*, *y*)**	**(*x*, *y*)**	**(*x*, *y*)**	**(*x*, *y*)**	**(*x*, *y*)**	**(*x*, *y*)**	**(*x*, *y*)**	**(*x*, *y*)**	**(*x*, *y*)**	**(*x*, *y*)**	**(*x*, *y*)**	**(*x*, *y*)**	**(*x*, *y*)**
Enlarge	-1, 0	-1, 0	-1, 0	-1, 0	-1, 0	-1, 0	0, 1	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0
Thinner	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	0, 1	-1, 0	-1, 0	-1, 0	-1, 0	-1, 0	-1, 0

Table III
MODIFICATIONS IN FIDUCIAL POINTS OF NOSE.

Fiducial points	**47**	**48**	**49**	**50**	**51**	**52**	**53**	**54**	**55**	**56**	**57**
Coordinates	**(*x*, *y*)**	**(*x*, *y*)**	**(*x*, *y*)**	**(*x*, *y*)**	(*x*, *y*)	**(*x*, *y*)**	**(*x*, *y*)**	**(*x*, *y*)**	**(*x*, *y*)**	**(*x*, *y*)**	**(*x*, *y*)**
Enlarge	-1, 0	-1, 0	-1, 0	-1, 0	-1, 0	0, 0	1, 0	1, 0	1, 0	1, 0	1, 0
Thinner	0, 0	0, -1	0, -1	0, 0	0, 1	0, 1	0, 1	0, 0	0, -1	0, -1	0, 0

Table IV
MODIFICATIONS IN FIDUCIAL POINTS OF CHIN.

Fiducial points	**4**	**5**	**6**	**7**	**8**
Coordinates	**(*x*, *y*)**	**(*x*, *y*)**	**(*x*, *y*)**	**(*x*, *y*)**	**(*x*, *y*)**
Triangular	0, 2	0, 3	0, 4	0, 3	0, 2
Squared	-2, 2	-2, 2	0, 0	2, 2	2, 2



(a) Change of initial coordinates. (b) Change of image dimensions.

Figure 7. Modifications in the coordinates and dimensions of detected faces.

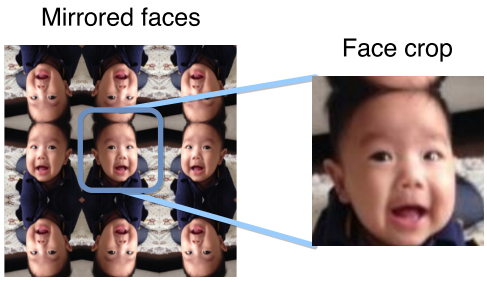


Figure 8. Example of face detected near the margin of the image [34].

V. EXPERIMENTAL EVALUATION AND RESULTS

The MORPH dataset [35], used for training the CNN face age classifier, consists of $55,132$ mug-shot images acquired from $13,618$ subjects and population age range is $16-69$ years old. That dataset is a multiethnic database, containing 77% of group Black and 19% of the group White, while the remaining 4% includes Hispanic, Asian, Indian, and Others. Data was acquired over a period of 5 years but not everybody provided samples every year. After the proposed data augmentation was applied, a total of $1,479,978$ images was generated. All deformations for a given source face kept the same label of the source. Some faces (318) not detected by the face detector were discarded from the training data. To the best of our knowledge, among exiting face datasets with age labels, MORPH presents the largest number of samples, less class unbalance (as opposed to FGRC [36]), and there is relevant published works using this dataset that allow comparative evaluations.

Scene images of MORPH dataset are similar with the images provided by the MUCT face dataset (e.g. they have similar brightness, backgrounds, and only frontal faces). This favored our detector of fiducial points, which presented an overall good matching scores with the age classifier. Available age range of the MORPH dataset (16 to 69 years) was split into n non-overlapping sub-ranges labeled i, $i = 1 \ldots n$. Two methods were used to evaluate the age classifier:

Method 1: The classifier output (estimated age) receives the label of the age sub-range into which it fits.

Method 2: Cumulative probability is calculated for each age sub-range. The classifier output (estimated age) receives the label of the age sub-range with highest cumulative probability.

In both cases the result is a hit (correct label assigned) or a miss (wrong label assigned). Both methods were measured by F-score [37]. F-score can be interpreted as a weighted average of the precision and recall, where an F-score reaches its best value at 1 and worst at 0.

Performance curves for both methods are shown in Figure 9, where GN-WDA and GN-WoDA are acronyms for GoogLeNet architecture training *With Data Augmentation* and *Without Data Augmentation*, respectively. GN-WoDA includes only images from MORPH dataset while GN-WDA includes the same images plus the faces generated by the proposed data augmentation approach. Indexes 1 and 2 indicate the evaluation method used.

Best performance was obtained by the GN-WDA1 combination, thus this was our choice to be used in further steps. The number of classes (Figure 9) represents the number of age sub-ranges (or intervals) into which the dataset age range (16 to 69 years) is evenly split. For example, if the number of classes is 4, there are 4 age intervals of 13 years ($16-28$, $29-41$, $42-54$ and $55-69$ years). Observe that, for 26 classes, the

GN-WoDA1 model obtains an F-score of 16%, and the GN-WDA1 model trained on the same dataset obtains an F-score of 26.1%, which is an increase of approximately 10% in F-score. For 3 classes, the GN-WDA model had an increase of 6.1% in the F-score. Although GoogLeNet is a state-of-the-art CNN for classification tasks, without our proposed data augmentation approach the obtained results were inferior to the results of the combined solution.

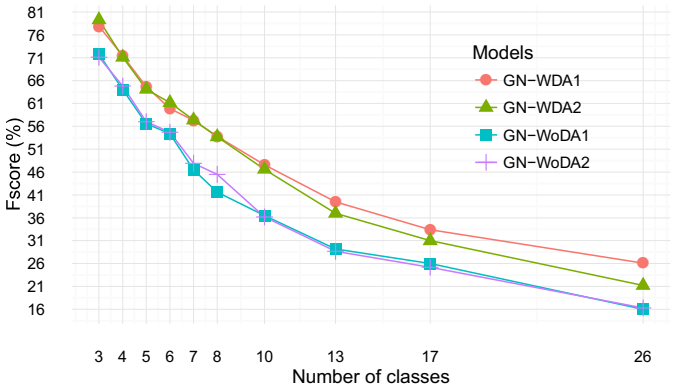


Figure 9. Performance of the two methods: with and without data augmentation.

The results presented in the work by Huerta et al.[13] and the best score presented in [14], for five different methods were used for comparison with our GN-WDA age estimator. In their work, a 5-fold cross-validation training was used on the MORPH database. The network was trained from scratch using the 50×50 MORPH dataset images in four folds and tested on the remaining one. Test results from the five possible assignments were averaged. Since we do not had access to the code of [13] and [14], the same evaluation procedure was adopted in our research in order to allow a comparative study. Corresponding performance curves are shown in Figure 10.

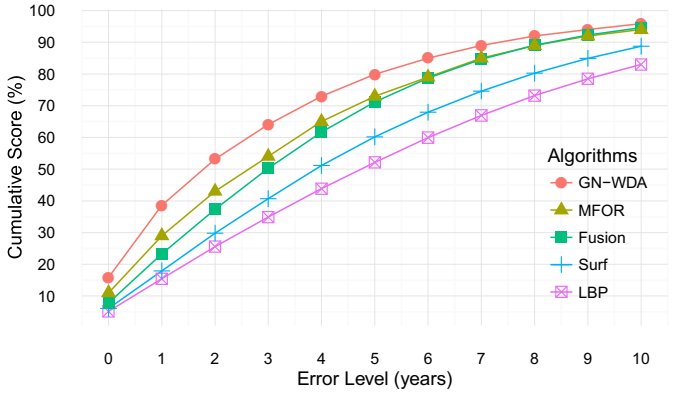


Figure 10. Comparison of different methods for age estimation. The CNN trained with proposed method of Data Augmentation is the top-scoring curve.

The vertical axis in Figure 10 represents the cumulative score, which is the percentage of images for which the age estimation error (difference between actual and predicted age, seen in the horizontal axis of Figure 10) is no higher than a given number of years (between 0 and 10) called of *error level* by [13]. The curves for the different techniques obtained from [13] and [14], can be compared with the one produced by CAFFE/GoogLeNet trained with the GN-WDA (augmented) MORPH dataset.

Parameter settings used in this research were the following: learning rate is 0.01, momentum is 0.9, weight decay is 0.0002 and batch size is 64. Output layer size was defined as 54 (to cover the 16 to 69 years old age interval).

Approximately 48 hours were needed to train one fold experiment using Nvidia Titan X GPU with 12GB of RAM. An input face image is classified in approximately 1s. Face fiducial point fitting, face deformations and disk operation took approximately 2s per image.

VI. Conclusions

The approach proposed and evaluated in this work aims at producing a more precise face age classifier by means of data augmentation. It is based on the detection of fiducial points in the face in order to generate smooth face image variations. Since the data influences in the learned information, a good way to improve the performance of recognition methods is to increase the dataset by means of data augmentation techniques, which can synthetically generate labeled data at a low cost.

A same CNN model (GoogLeNet) and face dataset (MORPH) were used to compare age prediction performance with and without the data augmentation approach proposed in this paper. Better results were observed when the classifier was trained with data augmentation. Moreover, competing face age prediction methods trained with the same database presented worse performance when compared with our results. A differential of the proposed approach is the fact that it takes into account information related to facial features to create new training samples, not simply lower level transformations associated with traditional data augmentation solutions.

As future work, more robust algorithms to detect fiducial points can be applied and other facial feature variations (e.g., eyes and mouth) could be incorporated. Moreover, additional deformations rules can be designed to allow for extra degrees of face variations and, thus, to check how far distortions and training set size can keep improving results. Other learning frameworks, such as TensorFlow [38] and CNTK [39], in addition to Caffe, could be explored. In order to show its generality, the proposed approach may be tested in other contexts, such as facial expression recognition and gender classification.

Acknowledgment

This work was developed as part of the Agreement 004/2015 between HP and UFCG, with funds from the return for Industrialized Product Tax (IPI) exemption or reduction granted by the Law 8.248, of 1991 and its subsequent updates.

REFERENCES

[1] S. Goyal and P. Benjamin, "Object recognition using deep neural networks: A survey," *CoRR*, vol. abs/1412.3684, 2014. [Online]. Available: http://arxiv.org/abs/1412.3684

[2] X. Cui, V. Goel, and B. Kingsbury, "Data augmentation for deep neural network acoustic modeling," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 9, pp. 1469–1477, Sept 2015.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[4] D. A. van Dyk and X.-L. Meng, "The art of data augmentation," *Journal of Computational and Graphical Statistics*, vol. 10, no. 1, pp. 1–50, 2001. [Online]. Available: http://www.jstor.org/stable/1391021

[5] A. Dosovitskiy, J. T. Springenberg, and T. Brox, "Unsupervised feature learning by augmenting single images," *CoRR*, vol. abs/1312.5242, 2013. [Online]. Available: http://arxiv.org/abs/1312.5242

[6] S. Gerke, K. Müller, and R. Schäfer, "Soccer jersey number recognition using convolutional neural networks," in *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, Dec 2015, pp. 734–741.

[7] X. Chen, Y. Xu, D. W. K. Wong, T. Y. Wong, and J. Liu, "Glaucoma detection based on deep convolutional neural network," in *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, Aug 2015, pp. 715–718.

[8] Y. Zhu, Y. Li, G. Mu, and G. Guo, "A study on apparent age estimation," in *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, Dec 2015, pp. 267–273.

[9] S. S. Farfade, M. J. Saberian, and L.-J. Li, "Multi-view face detection using deep convolutional neural networks," in *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, ser. ICMR '15. ACM, 2015, pp. 643–650.

[10] V. Jain and E. Learned-Miller, "Fddb: A benchmark for face detection in unconstrained settings," University of Massachusetts, Amherst, Tech. Rep. UM-CS-2010-009, 2010.

[11] N. McLaughlin, J. Martinez Del Rincon, and P. Miller, "Data-augmentation for reducing dataset bias in person re-identification," in *Advanced Video and Signal Based Surveillance (AVSS), 2015 12th IEEE International Conference on*, Aug 2015, pp. 1–6.

[12] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov 1998.

[13] I. Huerta, C. Fernández, C. Segura, J. Hernando, and A. Prati, "A deep analysis on age estimation," *Pattern Recognition Letters*, vol. 68, Part 2, pp. 239 – 249, 2015.

[14] C. Fernández, I. Huerta, and A. Prati, *Face and Facial Expression Recognition from Real World Videos: International Workshop, Stockholm, Sweden, August 24, 2014, Revised Selected Papers*. Cham: Springer International Publishing, 2015, ch. A Comparative Evaluation of Regression Learning Algorithms for Facial Age Estimation, pp. 133–144.

[15] R. Wang and Z. Xu, "A pedestrian and vehicle rapid identification model based on convolutional neural network," in *Proceedings of the 7th International Conference on Internet Multimedia Computing and Service*, ser. ICIMCS '15. New York, NY, USA: ACM, 2015, pp. 32:1–32:4.

[16] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the ACM International Conference on Multimedia*. ACM, 2014, pp. 675–678.

[17] R. Collobert, K. Kavukcuoglu, and C. Farabet, "Torch7: A matlab-like environment for machine learning," in *BigLearn, NIPS Workshop*, no. EPFL-CONF-192376, 2011.

[18] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, "Theano: A CPU and GPU math compiler in Python," in *Proc. 9th Python in Science Conf*, 2010, pp. 1–7.

[19] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going Deeper with Convolutions," *CoRR*, vol. abs/1409.4842, 2014. [Online]. Available: http://arxiv.org/abs/1409.4842

[20] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li, "ImageNet Large Scale Visual Recognition Challenge," *CoRR*, vol. abs/1409.0575, 2014. [Online]. Available: http://arxiv.org/abs/1409.0575

[21] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[22] M. Castelluccio, G. Poggi, C. Sansone, and L. Verdoliva, "Land use classification in remote sensing images by convolutional neural networks," *CoRR*, vol. abs/1508.00092, 2015. [Online]. Available: http://arxiv.org/abs/1508.00092

[23] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 681–685, Jun. 2001. [Online]. Available: http://dx.doi.org/10.1109/34.927467

[24] I. Matthews and S. Baker, "Active appearance models revisited," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 135–164, 2004.

[25] M. H. Mahoor, S. Cadavid, D. S. Messinger, and J. F. Cohn, "A framework for automated measurement of the intensity of non-posed facial action units," in *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*. IEEE, 2009, pp. 74–80.

[26] T. F. Cootes, C. J. Taylor *et al.*, "Statistical models of appearance for computer vision," University of Manchester, Tech. Rep., 2004.

[27] E. Antonakos, J. Alabort-i Medina, G. Tzimiropoulos, and S. Zafeiriou, "Hog active appearance models," in *Image Processing (ICIP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 224–228.

[28] T. F. Cootes, G. J. Edwards, C. J. Taylor *et al.*, "Comparing Active Shape Models with Active Appearance Models." in *BMVC*, vol. 99, no. 1, 1999, pp. 173–182.

[29] S. Milborrow, J. Morkel, and F. Nicolls, "The MUCT Landmarked Face Database," *Pattern Recognition Association of South Africa*, 2010.

[30] S. Schaefer, T. McPhail, and J. Warren, "Image deformation using moving least squares," in *ACM transactions on graphics (TOG)*, vol. 25, no. 3. ACM, 2006, pp. 533–540.

[31] S. Hosoi, E. Takikawa, and M. Kawade, "Ethnicity estimation with facial images," in *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, May 2004, pp. 195–200.

[32] N. Markus, M. Frljak, I. S. Pandzic, J. Ahlberg, and R. Forchheimer, "A method for object detection based on pixel intensity comparisons," *CoRR*, vol. abs/1305.4537, 2013. [Online]. Available: http://arxiv.org/abs/1305.4537

[33] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, 2001, pp. I–511–I–518 vol.1.

[34] E. Eidinger, R. Enbar, and T. Hassner, "Age and gender estimation of unfiltered faces," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 12, pp. 2170–2179, Dec 2014.

[35] K. Ricanek and T. Tesafaye, "MORPH: a longitudinal image database of normal adult age-progression," in *Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on*, April 2006, pp. 341–345.

[36] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek, "Overview of the face recognition grand challenge," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1. IEEE, 2005, pp. 947–954. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2005.268

[37] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009. [Online]. Available: http://dx.doi.org/10.1016/j.ipm.2009.03.002

[38] Martín Abadi et al., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," *CoRR*, vol. abs/1603.04467, 2016. [Online]. Available: http://arxiv.org/abs/1603.04467

[39] Amit Agarwal et al., "An Introduction to Computational Networks and the Computational Network Toolkit," Microsoft, Tech. Rep. MSR-TR-2014-112, August 2014. [Online]. Available: http://research.microsoft.com/apps/pubs/default.aspx?id=226641