CASCADED NONLINEAR SHAPE MODEL REGRESSION

Pedro Martins[1], *Bruno Silva*[1], *Jorge Batista*[2,1]

[1]Institute of Systems and Robotics, University of Coimbra, Portugal
[2]Department of Electrical and Computer Engineering, University of Coimbra, Portugal

ABSTRACT

This paper targets deformable face model matching in images using cascaded regression techniques. Recently, the cascaded regression strategies have became rather popular solutions to solve nonlinear objective functions by learning a pipeline sequence of linear regressors. However, despite their success, the standard formulation do not enforce shape consistency through the cascade (mostly because the individual regressors are learnt independently). In this paper we revisit the cascaded regression framework and propose a number of improvements. First we explore the simplicity and compactness of using a linear shape model for such tasks, effectively solving the previous drawback. Then we propose to extend the linear regression module into a nonlinear version, by means of recent Convolutional Neutral Networks (CNNs) techniques, modified to include a weighted shape aware loss function. Moreover, since CNNs often require massive amounts of data to perform well, we took advantage of the shape model probabilistic framework to efficiently bootstrap new data. Our nonlinear cascade regression method is evaluated in several databases (LFPW, LFW, HELEN and 300W), where the results demonstrate the effectiveness of the proposed method.

Index Terms— Non-rigid face registration, face alignment, deformable models, facial feature localization, cascaded regression.

1. INTRODUCTION

Nonrigid face registration, commonly known as face alignment, is the computer vision task that aims to locate a set of semantic facial features (landmarks) such as eyes, nose, mouth, etc. Such task is the at core of a number of applications, p.e. face recognition, face verification, pose estimation, video compression, etc.

During years, the Active Shape Model (ASM) [1] and afterwards the Active Appearance Model (AAM) [2] [3] were the gold standard for face alignment tasks. In fact, these approaches have popularized the usage of a Point Distribution Model (PDM) as a simple and efficient strategy to represent the spatial configuration of facial landmarks. The PDM is a linear compact model that represents the statistical modes of geometric variation of the training examples. The AAM extended the ASM by further include a generative appearance model combined with an image warping normalization step (piecewise affine warp). Matching (most times called fitting) an AAM into a target image, can be posed as a nonlinear optimization that finds the 'best' set of shape and appearance parameters that minimizes the objective function (typically, a L2 loss is involved). The AAM are intrinsically generative models. Meaning that it can exhibit a poor performance in unseen data, nevertheless some extensions were proposed to mitigate with this effect, such as Adaptive AAM [4] [5].

Discriminative appearance extensions appeared later, most notably with the introduction of the Constrained Local Model (CLM) [6] [7] [8] [9] [10] [11]. The CLM still retains the linear shape model but only accounts with the appearance of local regions (around each landmark). In fact, CLM uses an ensemble of local feature detectors (trained discriminatively) to scan locally, producing response maps. According, the CLM optimization seeks to find the set of shape parameters that maximize all the local detections at once.

Nowadays, the cascaded regression techniques [12] [13] [14] [15] [16] [17] [18] had became the mainstream approaches to face alignment. The cascaded regression framework uses a sequence of (linear) regressions in order to approximate an intricate mapping between image data and the landmark updates (in essence, converting a difficult regression problem in a summation of simple ones). In this setting, a bank of simple regressors is learnt offline from generated data (typically a large collection of virtual samples). Usually this process is computationally costly and high dimensional regressions are involved. In the other hand, fitting a cascaded model is a very efficient procedure, it simply consists of applying recursively each regressor (given an estimate) and keep collecting the shape updates. Mainly, these methods differ from each other by the way as the regression is accomplished, p.e. boosted regression [12] [19] [14] [15], least-squares regression [13] [18] or Gaussian Processes regression [16]. It is worth mentioning that there are other approaches that apply the same cascaded regression principle, not to image data directly but to a specific cost function, like AAM fitting [20] [21].

Recently, Convolutional Neural Networks (CNNs) techniques have emerged in virtually every computer vision task. In contrast with the previous methods, that require the use of suboptimal hand crafted features (such as HoG [22] or SIFT [23]), the CNNs have de ability to learn their own (strong) representations. Several face alignment CNN based approaches were proposed, we highlight just a few: The DCNN [24] is a standard CNN based regression approach was used to locate facial landmarks directly. Later, in [25] successive stacked auto-encoders were arranged in a coarse-to-fine strategy. The Mnemonic Descent Method (MDM) [26] extends SDM by combining CNNs and Recurrent Neural Networks. Deep learning extensions of CLMs and Deformable Parts Models (DPM) [27] have also been proposed in [28] and [29], respectively, where the first, referred as the Convolutional Experts Constrained Local Model (CE-CLM) [28], combines several CNNs structures per landmark (acting as multiple local detectors) and a global linear shape regularization.

In this paper we revisit the cascade regression framework and propose an nonlinear extension. Here we adopt a CNN strategy as base regressor, enhanced by a shape aware loss function. Additionally, we perform regression using a linear shape model, which encodes the shape constraint within the structure of the regression and promotes a reduced task effort. Moreover, a probabilistic shape model bootstrap strategy was used to consistently augment data to train the network. In the process, we leverage the image normalization process (warping), common in the AAM/CLM frameworks, to operate as a 'pose-free' canonical reference and use partial image observations to deal with mild occlusions.

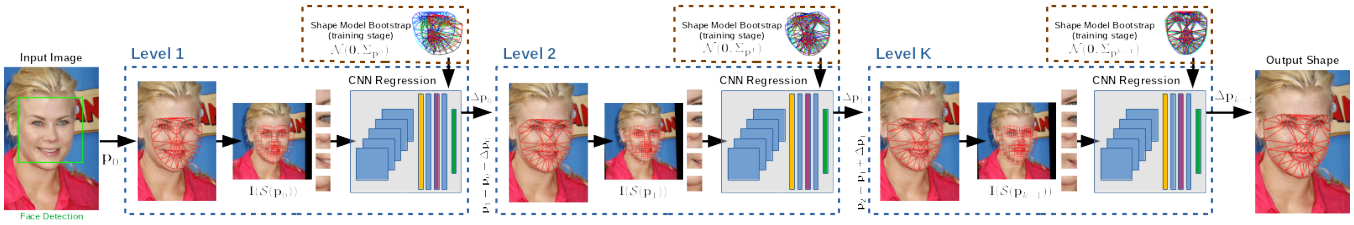


Fig. 1. Overview of the proposed nonlinear cascade regression formulation. Each cascade level has an image normalization step (similarity warp), local region sampling and a nonlinear regression stage (CNN) that provides the update to the shape model's parameters.

2. BACKGROUND

This section briefly reviews the basics of the cascaded regression as well as the linear shape model, widely used in deformable models.

2.1. Cascaded Linear Regression

Let us start by defining that a 2D shape, holding v landmarks, is represented by a vector $\mathbf{s} = (x_1, \ldots, x_v, y_1, \ldots, y_v)^T \in \mathbb{R}^{2v}$.

The key idea behind the cascade regression framework is to learn a sequence of K linear regressors and their bias $\{\mathbf{R}^k, \mathbf{b}^k\}_1^K$ that allow to approximate an nonlinear mapping between an initial shape (usually a mean shape adjusted to the output of a face detector) and its true location (the annotation ground truth),

$$\mathbf{s}^k = \mathbf{s}^{k-1} + \mathbf{R}^{k-1}\mathcal{F}(\mathbf{I}, \mathbf{s}^{k-1}) + \mathbf{b}^{k-1} \tag{1}$$

where the superscript index k represents a cascade level and $\mathcal{F}(\mathbf{I}, \mathbf{s}^{k-1}) \in \mathbb{R}^d$ denotes feature extraction in the input image $\mathbf{I}$ at the (previous) shape location $\mathbf{s}^{k-1}$. Feature extraction involves to sample v local regions, with a $P \times P$ support size centred at landmark $s_j = (x_j, y_j)$ and concatenate the results into a d sized vector. Each regression matrix $\mathbf{R}^k \in \mathbb{R}^{2v \times d}$ relates the extracted features to the additive updates to be made to the current shape estimate.

Learning each regression matrix can be obtained by minimizing the expected loss between the predicted and the optimal shape displacement under many possible initializations [30]

$$\arg\min_{\mathbf{R}^k} \sum_{I_l}^{N} \int p(\mathbf{s}_l^k) \left(\mathbf{s}_* - \left(\mathbf{s}_l^k + \mathbf{R}^k \mathcal{F}(\mathbf{I}, \mathbf{s}_l^k)\right)\right)^2 \partial \mathbf{s}_l^k \tag{2}$$

where N is the number of training images. Here, for sake of clarity, the bias term ($\mathbf{b}^k$) was omitted because it can be absolved into an additional column of the regression matrix ($\mathbf{R}^k \in \mathbb{R}^{2v \times d+1}$).

Assuming that $\mathbf{s}_l^k \sim \mathcal{N}(\mu_\mathbf{s}, \Sigma_\mathbf{s})$ is drawn from a Normal distribution (capturing the variance of the initial estimate provided by the face detector), the previous optimization 2 can be approximated by the discrete form

$$\mathbf{R}^k = \arg\min_{\mathbf{R}^k} \sum_{i=1}^{N} \sum_{j=1}^{M} \|\Delta \mathbf{s}_j^k - \mathbf{R}^k \mathcal{F}(\mathbf{I}_i, \mathbf{s}_j^k)\|^2 \tag{3}$$

where M is the number of perturbations / trial simulations and $\Delta \mathbf{s}_j^k = \mathbf{s}_* - \mathbf{s}_j^k$ is the difference between the ground truth ($\mathbf{s}_*$) and the disturbed shape. Note that the term $\Delta \mathbf{s}$ acts as regression labels. The least squares solution of eq. 3 takes the form of

$$\mathbf{R}^k = \Delta \mathbf{S} \left(\mathbf{F}^T \mathbf{F}\right)^{-1} \mathbf{F}^T \tag{4}$$

where $\mathbf{F}$ is a large data matrix holding all accumulated extracted features and $\Delta \mathbf{S}$ contains the corresponding shape deviations $\Delta \mathbf{s}_j$ for each of the M trials.

2.2. Linear Shape Model

As pointed out, we follow a PDM [31] like model. Briefly, this kind of model is captured from a set shape examples (annotations) where a Procrustes analysis is applied in each example, removing the similarity effects. Follows a Principal Components Analysis (PCA), which results in

$$\mathcal{B}(\mathbf{s}; \mathbf{b}) = \mathbf{s}_0 + \sum_{i=1}^{n} \phi_i b_i = \mathbf{s}_0 + \Phi \mathbf{b}, \quad \mathbf{b} \sim \mathcal{N}(\mathbf{0}, \Sigma_\mathbf{b}) \tag{5}$$

where $\mathbf{b} \in \mathbb{R}^n$ is the shape parameters vector (representing the deformation weights) and $\Phi \in \mathbb{R}^{2v \times n}$ is the shape subspace. The $\Sigma_\mathbf{b} = \text{diag}(\lambda_1, \ldots, \lambda_n)$ is covariance of the shape parameters with λ_i being the i^{th} data eigenvalue (provided by PCA).

Dealing with 2D pose is more troublesome. Fortunately we can avoid nonlinear equations in the similarity transformation by using the following reparameterization of the pose parameters $\mathbf{q} = [s\cos(\theta) - 1, s\sin(\theta), t_x, t_y]^T$ where s is the scale, θ the rotation and (t_x, t_y) are 2D translations. Under this setting [3], 2D pose function can be expressed as

$$\mathcal{S}(\mathbf{s}; \mathbf{q}) = \mathbf{s} + \sum_{j=1}^{4} \psi_j q_j = \mathbf{s} + \Psi(\mathbf{s})\mathbf{q}, \quad \mathbf{q} \sim \mathcal{N}(\mathbf{0}, \Sigma_\mathbf{q}) \tag{6}$$

note however that $\Psi(\mathbf{s}) \in \mathbb{R}^{2v \times 4}$ is a function of $\mathbf{s}$. The quantity $\Sigma_\mathbf{q}$ is the covariance of the pose parameters which can be estimated in the Procrustes procedure. For the sake of notation, we define a combined model having all the parameters together $\mathbf{p} \in \mathbb{R}^{n+4}$ (but keep in mind that $\mathbf{s} = \mathcal{S}(\mathcal{B}(\mathbf{b}); \mathbf{q})$), likewise $\Sigma_\mathbf{p}$ stands for the combined covariance matrix. Additionally we define $\mathbf{I}(\mathcal{S}(\mathbf{p}))$ to be the similarity (back) warped image $\mathbf{I}$ with the given (pose) parameters.

3. ENHANCED CASCADED REGRESSION

One major drawback of the cascade regression framework is that, in its basic form (as described in section 2.1), it does not enforce shape consistency [32] [26]. In this work we propose to extend this framework by exploiting the compactness of a linear shape model combined with the convenience of the pose representation, simultaneously taking advantage of nonlinear regression steps using a Convolutional Network Network (CNN).

Formally, our cascade follows the following recursion

$$\mathbf{p}^k = \mathbf{p}^{k-1} + \gamma \boldsymbol{\mathcal{R}}^{k-1}\{\mathcal{L}(\mathbf{I}(\mathcal{S}(\mathbf{p}^{k-1})))\} \tag{7}$$

where $\mathbf{p}$ are combined shape and pose parameters, $\boldsymbol{\mathcal{R}}^k$ is a nonlinear mapping function that regresses a set of local patches, taken from a 'pose-normalized' warped image, to the shape parameters and

$\mathcal{L}(.)$ is an operator that defines local image sampling (around each landmark). Finally, γ is the step-size which is included to minimize overshooting (found by cross-validation).

Figure 1 shows an overview of the proposed technique. Assuming that $\boldsymbol{\mathcal{R}}^k$ is known, the process of face alignment is accomplished by following eq. 7. Essentially, each cascade step k iterates between few stages: face normalization (image warp), local patch sampling, regression and, lastly, the parameters update. Given an estimate $\mathbf{p}_0$ (provided by a face detector), the input image is first similarity warped with the current pose estimate $\mathbf{I}(\mathcal{S}(\mathbf{p}^0))$. Follows local image sampling, where we choose to use a partial image representation to better deal with occlusions. The nonrigid shape parameters component provide the shape location, at the warped image reference, where we sample all the v local regions $\mathcal{L}(\mathbf{I}(\mathcal{S}(\mathbf{p}^0)))$. We gather a 3D array ($P \times P \times v$) of data that is feed as the input of a CNN (that effectively operates as being $\boldsymbol{\mathcal{R}}^k$) and perform the nonlinear regression, providing the parameters update to the next cascade stage.

3.1. Nonlinear Regression

As described before, a CNN structure is explored to perform nonlinear regression. In particular, we aim to estimate a nonlinear function $\boldsymbol{\mathcal{R}}^k = \{r_1, r_2, ...r_L\}$ with L layers for each step of the cascade k as

$$\boldsymbol{\mathcal{R}}^k = \arg\min_{\boldsymbol{\mathcal{R}}^k} \sum_{i=1}^{N}\sum_{j=1}^{M} \|\Delta\mathbf{p}_j^k - r_L(...r_1(\mathcal{L}(\mathbf{I}_i(\mathcal{S}(\mathbf{p}_j^k)))))\|^2_{\Sigma_{\mathbf{p}^k}} \tag{8}$$

where $\Delta\mathbf{p}_j^k = \mathbf{p}_* - \mathbf{p}_j^k$ is the shape parameters deviation from the ground truth (regression labels), M is again the number of virtual samples and

$$r_l(a_l) = \sigma(\mathbf{w}_l a_l + b_l). \tag{9}$$

In the previous eq. 9, the r_l is the mapping function of the l^{th} layer of the network, $\sigma(.)$ is the activation function, $\mathbf{w}_l$ is the weights (or filters) and a_l is the feature representation at each layer.

3.1.1. CNN Topology

The figure 2 shows a diagram with the CNN network topology. In essence, the network contains three convolutional layers dedicated to feature extraction, however, the first convolutional layer is deep-wise (forcing the first stack of filters to be specialized in each landmark independently). We recall that, the input of the network is a 3D array with size $P \times P \times v$. All convolutional layers are defined to have 32 filters, each with a kernel 3×3 and no padding. Each convolutional layer is followed by batch normalization and a rectified linear (ReLU) unit. Follows a dropout layer (0.4) to minimize overfitting, a fully connected layer and finally a regression layer to the $n + 4$ parameters. Finally, we highlight that the same CNN architecture is used across all the cascade stages.

3.1.2. Loss function

The optimization in 8 is defined in terms of a quadratic weighted error term. According, our CNN optimization uses the following loss function and gradient w.r.t. the regression predictions, respectively:

$$L_r = \frac{1}{N}\sum_{j=1}^{N} \Delta\mathbf{p}_j^T \Sigma_{\mathbf{p}}^{-1} \Delta\mathbf{p}_j, \quad \frac{\partial L_r}{\partial \mathbf{p}} = 2\Delta\mathbf{p}^T\Sigma_{\mathbf{p}}^{-1}. \tag{10}$$

The previous eqs. 10 were inspired by the Mahalanobis metric where each shape model dimension is weighted properly. It is worth mentioning that the cascade superscript k was omitted, but keep in mind that the covariance $\Sigma_{\mathbf{p}}$ shrinks across cascade levels.

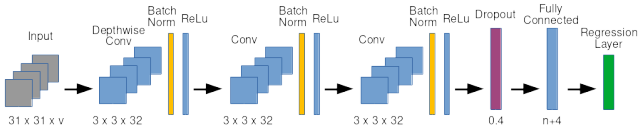


Fig. 2. Topology of the CNN in each stage of the cascade.

Learn the shape model $(\mathbf{s}_0, \Phi, \Psi)$
Get an initial estimate for all virtual instances $\mathbf{p}_{ij}^0$
for *cascade* $k = 1$ **to** K **do**
  $\mathbf{D} = \mathbf{0}_{P\times P\times v\times(N\times M)}$ //4D data buffer
  $\Sigma_{\mathbf{p}^k} = \text{cov}(\mathbf{p}_{ij}^k - \mathbf{p}_*)$ //estimate noise
  for *image* $i = 1$ **to** N **do**
    for *virtual sample* $j = 1$ **to** M **do**
      $\mathbf{p}_{ij}^k = \mathbf{p}_{ij}^k + \nu, \quad \nu \sim \mathcal{N}(\mathbf{0}, \Sigma_{\mathbf{p}^k})$ //add noise
      $\Delta\mathbf{p}_{ij}^k = \mathbf{p}_{ij}^k - \mathbf{p}_*$ //regression labels
      $\mathbf{I}_i(\mathcal{S}(\mathbf{p}_{ij}^k)) \leftarrow \mathbf{I}_i$ //warp image
      $\mathbf{D}_{ij} = \mathcal{L}(\mathbf{I}_i(\mathcal{S}(\mathbf{p}_{ij}^k)))$ //local regions
    end
    Estimate $\boldsymbol{\mathcal{R}}^k$ given $\{\Delta\mathbf{p}, \mathbf{D}\}$ //optimize CNN
  end
  $\mathbf{p}_{ij}^{k+1} \leftarrow \mathbf{p}_{ij}^k + \gamma\boldsymbol{\mathcal{R}}^k\{\mathbf{D}_{ij}\}$ //update cascade
end

Algorithm 1: Shape Model Regression (SMR) learning steps.

3.2. Virtual Samples Generation

A known issue in CNN approaches is that they require massive amounts of data to perform well. Here we propose to generate the M virtual samples (eq. 8) by bootstrapping the shape model. We can take advantage of the PDM probabilistic framework, combined with the image warping mechanism, to generate a virtually infinite number of samples, i.e. the i^{th} 'real' example can be augmented M times by

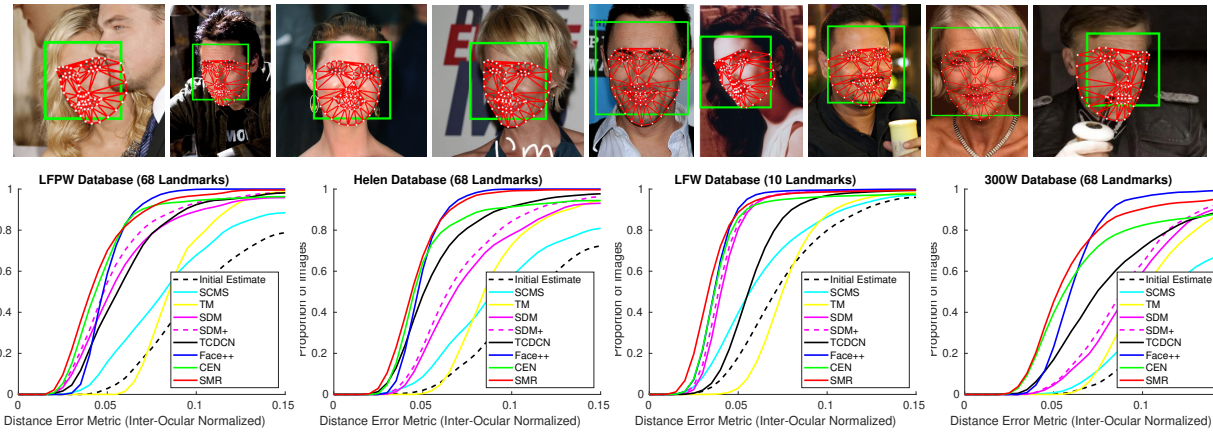
$$\mathbf{I}_i(\mathcal{S}(\mathbf{p}_i)) \longrightarrow \mathbf{I}_i(\mathcal{S}(\mathbf{p}_{ij})), \quad \mathbf{p}_{ij} \sim \mathcal{N}(\mathbf{p}_i, \Sigma_{\mathbf{p}^k}). \tag{11}$$

Data augmentation is done by generating perturbed shapes and poses simultaneously. The learning procedure is described in algorithm 1.

4. EXPERIMENTAL EVALUATION

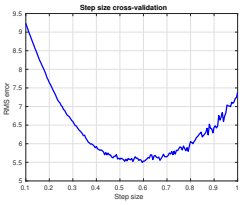
The evaluation takes place in several 'in the wild' databases. These datasets contain images that were acquired in unconstrained scenarios, i.e. under variations of lighting, focus, facial expression, pose and occlusions. A total of four datasets were used, namely: (1) The LFPW [37] database that has 811 (train) and 224 (test) images collected over web searches (68 landmarks [38]); (2) The HELEN [39] database holds 2000 (train) plus 330 (test) images taken from the flickr site (68 landmarks [38]); (3) The LFW [40], the largest set, has more than 13000 images (10 landmarks); The train/test portions had a split of 70/30; Finally, (4) the 300W [41] [38] consists of 600 images taken in both indoor and outdoor scenarios. The train set has a total of 6197 images taken from other datasets (68 landmarks).

The main evaluation compares our proposed technique, designated here as Shape Model Regression (SMR), against a classical Constrained Local Model (CLM) method (serving as a baseline), the part-based Tree-Model (TM) [27], standard cascaded regression techniques and also CNNs based techniques. The classical CLM is the Subspace Constrained Mean-Shifts (SCMS) [8]. The standard



(a) (b) (c) (d)

(%) Area under fitting curve	LFPW	HELEN	LFW	300W
Initial Estimate [33]	30.5	25.4	49.0	17.6
SCMS [8]	42.4	36.0	57.3	23.2
TM [27] (p146)	40.8	39.7	47.7	30.6
SDM [13]	60.2	48.6	71.5	36.5
SDM+ (w/ linear shape model)	63.6	52.1	72.4	39.0
TCDCN [34]	59.5	61.1	61.0	44.6
Face++ [35] [36]	66.7	67.4	74.8	58.4
CEN [28]	67.2	63.6	72.3	54.0
SMR (our method)	69.6	69.1	75.9	59.5



(e)

Computational performance.

	Learning	Execution
SMR*	180 min / cascade	34 ms / cascade

(*) Intel i7-3930K, 32GB, Nvidia GTX Titan X.

Fig. 3. Fitting performance curves in the (a) LFPW, (b) HELEN, (c) LFW and (d) 300W databases, respectively. The table show a quantitative measure of area under the fitting curve. (e) Step size cross-validation. The images on top are SMR fitting examples in the LFPW dataset.

cascaded regression approaches are the Supervised Descent Method (SDM) [13], which was described in section 2.1 and a variation on this, referred to as SDM+, which corresponds to an implementation that makes regression using a linear shape model. The CNNs based techniques are the Tasks-Constrained Deep Convolutional Network (TCDCN) [34], the Convolutional Experts CLM (CE-CLM) [28] and finally, the commercial solution Face++ [35] [36].

The methods SCMS, SDM, SDM+ consist of our own implementations. The remaining techniques are based in each author's supplied code (with the Face++ providing a free developer API). All methods were initialized with the mean shape and the pose parameters obtained by regression (from the output of a face detector [33] to its corresponding parameters). In SCMS, the local landmark detectors have a support size of 31×31, being MOSSE filters [42] built from grey level intensities. The individual response maps optimization include mean-shift updates with a kernel bandwidth schedule of $(15, 10, 5, 2)$. Both the cascaded regression models (SDM and SDM+) use HoG [22] features, have a local path size of 33×33 (cell size = 3) and run for $K = 6$ cascade levels.

Regarding our SMR approach, each image is sampled around each landmark using a local region of 31×31 (where we remove the mean and normalize by the standard deviation each region individually), the shape model holds 97.5% of the variance of the data (resulting in $n = 27$ shape parameters in the LFPW dataset). Each CNN regression network was optimized with Adam [43] using the default hyperparameters, an initial learning rate of 10^{-4} with exponential decay of 0.90 every 5000 iterations and a mini-batch with 64 examples. The number of cascade levels was set to $K = 6$.

The cascade recursion step-size (γ) was subject to an additional cross-validation to find a suitable value. The figure 3-(e) shows the average Root Mean Squared error in a randomly selected test set (500 images taken from LFPW and HELEN) for several values of γ. The minimal value of $\gamma = 0.595$ was found and used afterwards.

The fitting performance is measured, as standard, by the normalized alignment error. Such a measure is given by the mean error per landmark as fraction of the inter-ocular distance, d_{eyes}, as $e_m(\mathbf{s}) = \frac{1}{v\ d_{\text{eyes}}} \sum_{i=1}^{v} \|\mathbf{s}^i - \mathbf{s}_*^i\|$ where $\mathbf{s}_*^i$ is the location of i^{th} landmark in the ground truth shape annotation. The figure 3 shows the cumulative error distribution functions for all the techniques in every dataset. The table included in the same figure shows a quantitative measure of the results which is defined as the ratio, in percentage, between the area bellow the fitting curve and the total area of a ground truth curve. The shape initialization is included in the evaluation charts. Additionally, SMR learning and execution times are shown.

The results show that the CLM method (SCMS) performs better that TM (whose limited accuracy comes from the simple regularization used, designed for fast inference), followed by the SDM cascaded regression methods. However, and as expected the CNN based solutions take the lead in terms of fitting accuracy. When comparing SDM with SDM+, we can see that there is a significant difference between regressing in x/y spatial locations and doing the same but with a constrained shape model (that has an underlying structure). The same logic can be applied to justify the difference in performance between Face++ and our SMR method. The CEN being a CLM technique includes a PDM, however, the CNN regression is landmark based, meaning that the shape model is just used as a spatial regularization. In contrast, our SMR approach embeds the global shape constraint within the structure of the regression.

5. CONCLUSIONS

This paper targets the cascaded regression framework with a nonlinear extension. Our proposed model takes advantage of a compact linear shape model combined with a series of CNNs, effectively encoding the shape within the structure of regression. The experimental results demonstrate the accuracy and performance of our method.

6. REFERENCES

[1] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models-their training and application," *CVIU*, vol. 61, no. 1, pp. 38–59, 1995.

[2] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE TPAMI*, vol. 23, no. 6, pp. 681–685, 2001.

[3] I. Matthews and S. Baker, "Active appearance models revisited," *IJCV*, vol. 60, no. 1, pp. 135–164, November 2004.

[4] A. U. Batur and M. H. Hayes, "Adaptive active appearance models," *IEEE TIP*, vol. 14, no. 11, pp. 1707–1721, 2005.

[5] T. F. Cootes and C. J. Taylor, "An algorithm for tuning an active appearance model to new data," in *BMVC*, 2006.

[6] D. Cristinacce and T. F. Cootes, "Automatic feature localisation with constrained local models," *PR*, vol. 41, no. 10, pp. 3054–3067, 2008.

[7] Y. Wang, S. Lucey, and J. Cohn, "Enforcing convexity for improved alignment with constrained local models," in *IEEE CVPR*, 2008.

[8] J. Saragih, S. Lucey, and J. Cohn, "Deformable model fitting by regularized landmark mean-shifts," *IJCV*, vol. 91, no. 2, pp. 200–215, 2010.

[9] T. F. Cootes, M. Ionita, C. Lindner, and P. Sauer, "Robust and accurate shape model fitting using random forest regression voting," in *ECCV*, 2012.

[10] P. Martins, J. F. Henriques, R. Caseiro, and J. Batista, "Bayesian constrained local models revisited," *IEEE TPAMI*, vol. 38, no. 4, pp. 704–716, April 2016.

[11] P. Martins, R. Caseiro, and J. Batista, "Non-parametric bayesian constrained local models," in *IEEE CVPR*, 2014.

[12] X. Cao, Y. Wei, F. Wen, and J. Sun, "Face alignment by explicit shape regression," in *IEEE CVPR*, 2012.

[13] X. Xiong and F. De la Torre, "Supervised descent method and its application to face alignment," in *IEEE CVPR*, 2013.

[14] X. P. Burgos-Artizzu, P. Perona, and P. Dollár, "Robust face landmark estimation under occlusion," in *IEEE ICCV*, 2013.

[15] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *IEEE CVPR*, 2014.

[16] D. Lee, H. Park, and C. D. Yoo, "Face alignment using cascade gaussian process regression trees," in *IEEE CVPR*, 2015.

[17] S. Zhu, C. Li, C. Loy, and X. Tang, "Face alignment by coarse-to-fine shape searching," in *IEEE CVPR*, 2015.

[18] E. S. Lozano, G. Tzimiropoulos, B. Martinez, F. De la Torre, and M. Valstar, "A functional regression approach to facial landmark tracking," *IEEE TPAMI*, 2018.

[19] S. Ren, X. Cao, Y. Wei, and J. Sun, "Face alignment at 3000 fps via regressing local binary features," in *IEEE CVPR*, 2014.

[20] G. Tzimiropoulos, "Project-out cascaded regression with an application to face alignment," in *IEEE CVPR*, 2015.

[21] P. Martins and J. Batista, "Simultaneous cascaded regression," in *IEEE ICIP*, 2018.

[22] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE CVPR*, 2005.

[23] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[24] Y. Sun, X. Wang, and X. Tang, "Deep convolutional network cascade for facial point detection," in *IEEE CVPR*, 2013.

[25] Jie Zhang, Shiguang Shan, Meina Kan, and Xilin Chen, "Coarse-to-fine auto-encoder networks (cfan) for real-time face alignment," in *ECCV*, 2014.

[26] G. Trigeorgis, P. Snape, M. Nicolaou, E. Antonakos, and S. Zafeiriou, "Mnemonic descent method: A recurrent process applied for end-to-end face alignment," in *IEEE CVPR*, 2016.

[27] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *IEEE CVPR*, 2012.

[28] A. Zadeh, T. Baltrušaitis, and L. P. Morency, "Convolutional experts constrained local model for facial landmark detection," in *IEEE CVPRW, 2nd Facial Landmark Localisation Competition*, 2017.

[29] K. Songsri-in, G. Trigeorgis, and S. Zafeiriou, "Deep & deformable: Convolutional mixtures of deformable part-based models," in *IEEE AFGR*, 2018.

[30] X. Xiong and F. De la Torre, "Supervised descent method for solving nonlinear least squares problems in computer vision," Tech. Rep. arXiv:1405.0601, 2014.

[31] T. F. Cootes and C. J. Taylor, "Statistical models of appearance for computer vision," Tech. Rep., Imaging Science and Biomedical Engineering, University of Manchester, 2004.

[32] E. Sánchez-Lozano, B. Martinez, and M. F. Valstar, "Cascaded regression with sparsified feature covariance matrix for facial landmark detection," *Pattern Recognition Letters*, 2016.

[33] P. Viola and M. Jones, "Robust real-time object detection," *IJCV*, vol. 57, no. 2, pp. 137–154, July 2002.

[34] Z. Zhang, P. Luo, C. Loy, and X. Tang, "Facial landmark detection by deep multi-task learning," in *ECCV*, 2014.

[35] Face++, "Megvii face api," 2018.

[36] Z. Huang, E. Zhou, and Z. Cao, "Coarse-to-fine face alignment with multi-scale local patch regression," arXiv:1511.04901, November 2015.

[37] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar, "Localizing parts of faces using a consensus of exemplars," in *IEEE CVPR*, 2011.

[38] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "300 faces in-the-wild challenge: The first facial landmark localization challenge," in *IEEE ICCV Workshop*, 2013.

[39] V. Le, J. Brandt, Z. Lin, L. Boudev, and T. S. Huang, "Interactive facial feature localization," in *ECCV*, 2012.

[40] G. B. Huang, M. Ramesh, T. Berg, and E. L. Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," Tech. Rep. 07-49, University of Massachusetts, Amherst, 2007.

[41] C. Sagonas, E. Antonakos, G. Tzimiropoulos, and M. Pantic, "300 faces in-the-wild challenge: database and results," *IVC, Special Issue on Facial Landmark Localisation 'In-The-Wild'*, vol. 47, pp. 3–18, 2016.

[42] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *IEEE CVPR*, 2010.

[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.