Accurate Regression Procedures for Active Appearance Models

Patrick Sauer
patrick.sauer@postgrad.man.ac.uk
Tim Cootes
tim.cootes@manchester.ac.uk
Chris Taylor
chris.taylor@manchester.ac.uk

Imaging Science and Biomedical Engineering
University of Manchester
Manchester, UK
M13 9PT

Abstract

Active Appearance Models (AAMs) are widely used to fit shape models to new images. Recently it has been demonstrated that non-linear regression methods and sequences of AAMs can significantly improve performance over the original linear formulation. In this paper we focus on the ability of a model trained on one dataset to generalise to other sets with different conditions. In particular we compare two non-linear, discriminative regression strategies for predicting shape updates, a boosting approach and variants of Random Forest regression. We investigate the use of these regression methods within a sequential model fitting framework, where each stage in the sequence consists of a shape model and a corresponding regression model. The performance of the framework is assessed by both testing on unseen data taken from within the training databases, as well as by investigating the more difficult task of generalising to unrelated datasets. We present results that show that (a) the generalisation performance of the Random Forest is superior to that of the linear or boosted regression procedure and that (b) using a simple feature selection procedure, the Random Forest can be made to be as efficient as the boosting procedure without significant reduction in accuracy.

1 Introduction

Active Appearance Models [4, 9] represent a well known group of algorithms for fitting shape models to images. Training the models requires a database of images in which a set of locations which characterise the object group in question have been labelled. In the original formulation in [9], the model was chosen to be linear and generative, i.e. an explicit model of the input data was provided. This enabled an iterative Gauss-Newton type procedure, where the error between the current image features and those synthesised using the current location of the model in the image was used to derive additive updates to the shape model parameters. However, such methods incur a substantial computational overhead, as an explicit model of the image features must be fitted and evaluated at each iteration of the algorithm. This efficiency problem was addressed by Baker et al. in [4], where an inverse compositional alignment procedure was proposed.

In related work, discriminative methods have shown promising results, both in terms of efficency and generalisability [10, 18, 20, 23]. Contrary to the generative approach, these methods do not include a model of the input data and therefore do not explicitly encode any information about the texture of the objects. Rather than using the Gauss-Newton style algorithm described above, discriminative fitting procedures involve a regression algorithm which directly relates features extracted from the image data to incremental updates of the internal parameters of the shape model.

A well-known problem of these algorithms is the often poor generalisation performance when attempting to fit to unseen data. In face analysis, this effect is especially pronounced when the lighting and background variations, as well as the people in the test dataset do not match those present in the training dataset. For this reason, much prior work has focused on evaluating the performance of Active Appearance Model algorithms on datasets which have been split into training and testing parts rather than on investigating the more challenging issue of employing unrelated datasets for training and testing.

Substantial effort has been put into addressing the generalisation performance issue within the generative framework, and notable examples are the Adaptive AAM [5] and multilinear models [16] which both aim to provide increased flexibility in the texture component of the model.

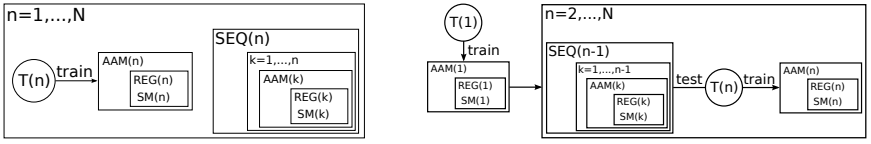
In this paper we compare two types of non-linear regression models in a discriminative framework, the first being based on Boosting [14], the second on our implementation of the Random Forest [7]. In order to account for the large variation in lighting conditions and appearance when comparing different datasets, we combine these algorithms with a set of standard Haar-like features [22] as well as illumination-independent features derived from the integral image, akin to the BRIEF features which were recently proposed by Calonder et al. in [8]. These features capture relative differences in pixel values accross images, making them invariant to monotonic transformations of the intensity. While the boosting-based method has been the subject of various publications in this field, we are unaware of any previous uses of the Random Forest in this particular setting.

The outline of the paper is as follows: We give a brief discussion of previous work in 1.1 before describing the methods employed in this paper in 2. The experiments are described in section 3, section 4 contains the results and conclusions are presented in 5.

1.1 Previous Work

The Boosting methods employed in this paper are closely related to the work of Zhou et al. [23] who used a regularised version of the Gradient Boost algorithm [13] to directly predict updates to a parametric shape model based on sampled image patches. Similarly, in [10] Cristinacce et al. compare the use of a GentleBoost classifier [14] to Boosted regression for fitting deformable models. In [20], Tresadern et al. investigate sequential boosted regression models for discriminative shape model fitting, where it is shown that substantial performance increases may be obtained by explicitly using the residual uncertainty after prediction with a coarse model as a starting point for an iterative training procedure that yields a sequence of increasingly refined models. Their work draws on previous work by Saragih et al. in [19] which uses a maximum-margin optimisation procedure to build a sequence of regression functions with guaranteed error bounds. Similarly, Dollár et al. use cascades of Random Fern regressors for object pose prediction in [11].

Rather than using the image data directly, Haar-like features [22] provide an efficient and powerful means for capturing facial features and their use has been described in several



(a) Independent Sequential AAM (b) Coupled Sequential AAM

Figure 1: Training procedures for the Sequential AAM models. **Left:** The Independent Sequential AAM consists of independently trained Regression AAM models. Each training dataset T(n) contains a prespecified range of displacements from the ground truth. **Right:** At each iteration of the training procedure, the Coupled Sequential AAM employs a training dataset obtained by testing the partial sequence SEQ(n-1) on dataset T(n), which contains random displacements of the same magnitude as T(1).

appearance model-related publications [10, 20]. Other choices which have been used previously include Local Binary Patterns [8], mutual information [12] and local eigenmodels [21].

2 Methods

2.1 Sequential Regression AAM

In regression-based Active Appearance Models, the model parameters are updated directly by applying the learned regression model to features extracted from the image at the current model location. This is more efficient than the generative fitting procedure, in which the shape parameter updates are obtained by iteratively minimising a suitably chosen error measure which relates the current texture model reconstruction to the current texture sampled from the image [17]. However, an iterative regression prediction lacks the reinforcement provided by the texture model during the optimisation in the generative framework. The accuracy of the regression procedure is therefore limited by the training data used to train the regression models. If large displacements from the ground truth are used to train the model, it can be expected to perform well when tested on examples showing similar displacements. However, when applied in an iterative fashion, the model is expected to move closer to the ground truth, and thus further from the type of examples provided in the training data, making the ensuing predictions approximate at best. One way of addressing this issue, termed the Sequential AAM in the following, builds on an idea first described by Saragih et al. in [19]. The Sequential Regression AAM trains a sequence of Regression AAM models of increasing complexity, where the models in the earlier stages typically only model pose variation, and shape variation is introduced in the later stages. We investigate two methods for training a Sequential AAM. In the first method which we refer to as the Coupled Sequential AAM, a training dataset containing model displacements of a prespecified magnitude is generated and used to train a Regression AAM. A separate dataset is then formed by gathering the predictions obtained by testing the partially trained sequence on a separate dataset sampled from the training images. Since the magnitude of the displacements in the training data decreases from stage to stage, and the complexity of the shape models increases, each stage in the sequence can be seen to represent a Regression AAM specialised to a certain range of displacements from the ground truth. A downside to this procedure, however, is the use of

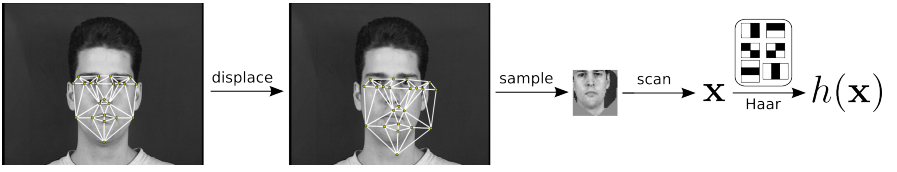


Figure 2: The training and testing data is generated by (a) fitting a shape model to the landmark points (b) displacing the parameters of the fitted model randomly (c) sampling into the reference frame of the shape model (d) raster-scanning into the vector $\mathbf{x}$ and computing Haar features and (f) recording the inverse displacements $\delta\mathbf{y}$ (not shown).

the outputs of the previous stages of the sequence as training data for the following stage, which may result in overfitting of the regression models to the training data. In order to assess the degree of overfitting, we propose an alternative training procedure we refer to as the Independent Sequential AAM, in which each stage in the sequence is trained independently, using training data whose range of displacements from the ground truth images is reduced by an empirically chosen factor from stage to stage.

Algorithm 1 Gradient Boost Training [13] for shape model regression [20]

1. Initialise:
   (a) $T = \{(\mathbf{x}_n, \delta\mathbf{y}_n)\}_{n=1}^{N}$
   (b) $F(\mathbf{x}) = 0$, $\lambda \ll 1$
2. Repeat for $m = 1,\ldots,M$:
   (a) Repeat for $i = 1,\ldots,d$:

$$(h_m, f_m) = \arg\min_{f,h} \sum_{n=1}^{N} \left[\delta\mathbf{y}_{n_i} - f_m(h_m(\mathbf{x}_n))\right]^2$$

   (b) update strong learner: $F(\mathbf{x}) \leftarrow F(\mathbf{x}) + \lambda\mathbf{f}_m(\mathbf{x})$
   (c) update targets: $\delta\mathbf{y}_n \leftarrow \delta\mathbf{y}_n - \lambda\mathbf{f}_m(\mathbf{x}_n)$
3. Output strong learner $F(\mathbf{x})$

2.2 Regression Models

2.2.1 Training data

In the following we shall refer to the data used to train the regression models as the set of pairs

$$T = \{(\mathbf{x}_1, \delta\mathbf{y}_1), \ldots, (\mathbf{x}_N, \delta\mathbf{y}_N)\}, \tag{1}$$

where $\mathbf{x}_n$ is the vector of pixels sampled from the reference frame of the model after randomly displacing the model from it's ground truth position and $\delta\mathbf{y}_n \in \mathbb{R}^d$ contains the shape and pose parameter displacements required to move the model back to the ground truth position. This procedure is illustrated in figure 2. The integral image is computed for every pixel sample $\mathbf{x}$, as this allows the efficient calculation of Haar-like features [22] which are used exclusively in the following.

2.3 Boosted regression

The Boosting algorithm follows the Gradient Boosting method [13], which additively combines a set of M *weak learners* f_m into a "strong" regression function, such that the update to the model parameters is given by

$$\delta y_i = \sum_{m=1}^{M} \lambda f_m(h_m(\mathbf{x})). \tag{2}$$

In this equation, $h_m(\mathbf{x})$ represents a Haar-like feature computed on the current pixel sample vector $\mathbf{x}$, and λ is a "shrinkage" parameter, which is included to reduce overfitting [15]. We explore two different kinds of weak learners f_m. The first are piecewise constant functions which are efficiently fitted in each cycle of the feature-finding procedure by binning the feature responses for each of the training examples and calculating the mean in each bin. If we are attempting to construct models capable of generalising to unseen data however, the use of less flexible functions may be advantageous, as unconstrained flexibility may lead to overfitting. In addition to the piecewise constant functions, we therefore investigate the use of simple tree stumps with a threshold fixed at zero. This is inspired by the efficiency and performance of BRIEF features [8], which build a feature representation of an image by performing simple binary comparisons between the mean of the pixels contained in different regions of interest. As in the case of BRIEF, the idea is that fixing the split at zero combined with Haar-like features produces a more generic, lighting-insensitive feature with superior generalisation performance. The pseudocode for the training stage of Gradient Boosting is given in figure 1.

2.4 Random forest regression

In recent years Random Forests [7] have become increasingly popular and have been used successfuly in many classification and regression problems. The popularity of Random Forests stems from the fact that they represent a simple and efficient algorithm with few free parameters and have been shown to resist overfitting in many applications. Furthermore, both the training and testing stages of Random Forests follow "embarassingly parallel" procedures, a fact which may be exploited to obtain high performance on parallel computer architectures. Random Forests are constructed by building a set of n binary trees on bootstrap samples of the training dataset T. In order to use Random Forests in 1D regression problems, the trees are constructed recursively, such that at each node the training data is split by choosing a threshold on a feature variable chosen at random from a subset of all the features that minimises the sum of squared errors

$$S_{sc} = \sum_{l \in L} \sum_{i \in l} (\delta y_i - m_l)^2 = \sum_{l \in L} n_l * var(l), \tag{3}$$

where $m_l = \frac{1}{m_l} \sum_{i=1}^{m_l} \delta y_i$ and $var(l)$ are the mean and variance of the samples δy_i contained in leaf l. Various stopping criteria for the recursion have been proposed in the literature [6]. In our case, the trees are built until each node contains a single sample. The trees are similar to the piecewise constant functions used as weak learners in section 2.3, as each leaf of a tree outputs the mean of the training data responses δy_i it contains. When presented with test data, the mean over the outputs of each tree in the forest is returned as the forest prediction. This ensemble prediction allows for the gathering of statistics on the outputs by inspecting the level of "agreement" among the individual tree regression functions.

The tree building procedure described above represents the standard procedure for scalar outputs δy_i. In order to allow for vectorial outputs $\delta \mathbf{y}_i$, we may either alter this criterion by extending the sum of squared errors to all output dimensions, i.e.

$$S_{vec} = \sum_{d=1}^{n_D} \sum_{l \in L} \sum_{i \in l} (\delta \mathbf{y}_{i_d} - \mathbf{m}_{l_d})^2 = \sum_{d=1}^{n_D} \sum_{l \in L} n_l * var_d(l), \tag{4}$$

such that at each binary split the feature that gives the best joint prediction of the outputs is chosen, or simply train a scalar Random Forest for each output dimension. Using the first method, we arrive at a regression forest capable of directly predicting vectorial outputs. This makes for a more efficient algorithm when compared to the set of scalar forests, and, depending on the dataset, the joint feature selection may also allow for important correlations in the outputs to be learned. However, in the case of uncorrelated outputs this could also prove to be a significant shortcoming, as especially for higher dimensional output vectors, a single feature is required to discriminate between possibly uncorrelated phenomena.

Apart from the standard thresholding procedure, as with the Boosting algorithm described above, we also investigate the use of fixed zero-thresholds in the tree-building process. Again, the idea is inspired by the BRIEF features described in [8], with the intention of producing a generic, lighting-insensitive feature with good generalisation performance on independent testing datasets.

3 Experiments

In this section we present experiments carried out on two datasets, XM2VTS [2] and BioID [1], for which ground truth positions of 22 key facial features are available as shown in figure 2. Both datasets contain frontal images of faces, but were captured under very different conditions. The XM2VTS dataset contains high-resolution images captured under controlled conditions, resulting in very little pose and lighting variation between different individuals. On the other hand, the BioID dataset consists of webcam images of people in offices. The images in BioID are of substantially lower quality and contain significant variation in pose and lighting. In our experiments we selected 400 images from both datasets, which were split into training and testing datasets containing 200 images. Pixel samples containing 1000 pixels were gathered from 10 random displacements of the landmark points in each image by up to 15% of the inter-ocular distance (IOD), such that $N = 2000$ training samples were used to train the regression models.

3.1 Experimental setup

In the experiments carried out in this paper, we insert the regression methods discussed in sections 2.3 and 2.4 into the Sequential Active Appearance Model framework introduced in section 2.1. In order to evaluate the performance of the different regression algorithms and feature selection criteria discussed in 2.2, we carried out a series of experiments in which a single Regression AAM model was used to predict the pose (i.e. scale, rotation and location) of the ground truth in the testing data.

Following up on this, a further experiment was carried out to compare the performance of Boosted regression and Random Forest regression, but this time involving 5-stage Sequential AAM models trained using the two procedures outlined in section 2.1. In the following, we

use two letter acronyms to describe the regression algorithm and features used, where 'pw' refers to Haar-like features with piecewise constant functions, 'vs' refers to the Random Forest using the multi-output variance split criterion and 'zs' is short for the zero-stump/zero-split feature selection procedures described in sections 2.3 and 2.4. The suffix '_1d' refers to the implementation of the Random Forest which uses a set of scalar trees as described in section 2.4. For completeness, we also include the results obtained using a simple linear regression function which was trained on the pixel data directly and which is denoted 'lin'.

In the Boosting algorithm, we chose to use 200 weak learners and a shrinkage parameter $\lambda = 0.05$. The Random Forests consist of 100 trees which were built until each leaf node contained a single sample.

4 Results

The errors after prediction are given as percentages of the inter-ocular distance (IOD). In tables 1, 2 and 3, we present the median of the displacement distributions before ('init') and after prediction using the models, as well as the percentage of searches that successfully converged. The predictions are obtained by running a single iteration of every model in the sequences. We consider a search to have converged successfully if the resulting points have a mean absolute displacement of 3% of the IOD from the ground truth. This is a fairly stringent threshold chosen to highlight the differences between the methods. In order to ensure consistency across datasets, the BioID images were resampled to the same resolution as the XM2VTS dataset (720×576).

4.1 Pose prediction

Table 1 shows the results of the pose prediction experiment. The results show a clear preference for the 1D implementation of the Random Forest. This is especially noticeable in the difficult case of training on the XM2VTS dataset and testing on BioID, where the 'vs_1d' algorithm has a success rate that is almost twice as large when compared to the Boosting algorithms 'pw' and 'zs'. Interestingly, the vectorial implementations of the Random Forest ('vs' and 'zs') perform significantly worse than their 1D counterparts, an indication that requiring a single feature to discriminate between the four pose dimensions is too constraining. This is especially clear in the case of the 'zs' Random Forest.

(a) test set: BioID

				boost		forest			
training set		init	lin	pw	zs	vs	vs_1d	zs	zs_1d
BioID	median	7.3	7.3	3.4	3.5	3.6	3.0	4.9	**2.9**
	success	5.0	3.0	43.0	36.9	37.5	51.3	19.8	**51.9**
XM2VTS	median	7.3	9.3	5.9	5.5	5.0	**3.9**	6.0	4.0
	success	5.0	2.7	17.7	15.6	20.3	**30.8**	9.6	30.7

(b) test set: XM2VTS

				boost		forest			
training set		init	lin	pw	zs	vs	vs_1d	zs	zs_1d
BioID	median	7.8	8.7	3.5	4.2	4.1	**3.0**	6.0	3.0
	success	4.0	1.0	36.7	25.9	32.5	**51.0**	13.5	49.8
XM2VTS	median	7.8	5.0	2.6	3.0	3.1	2.2	5.3	**2.2**
	success	4.0	11.6	60.3	48.7	48.8	**67.6**	18.5	67.1

Table 1: Median of the displacement distribution and percentage of successfully converged samples obtained by predicting pose updates only using one iteration of a single Regression AAM. The two letter acronyms describe the different algorithms and are explained in the text. **Left:** Testing performed on the BioID database. **Right:** Testing performed on the XM2VTS database.

4.2 Sequential AAM

From the results in tables 2 and 3, it may be inferred that the coupled sequential training algorithm leads to overfitting of the training data. This becomes clear when comparing the performance in the case of training on XM2VTS and testing on BioID, where the independent algorithm significantly outperforms the coupled algorithm in all but the case of 'zs'. Interestingly, the Boosting algorithms appear to outperform their Random Forest counterparts in the coupled case, although their performance is significantly worse when compared to the best results obtained using the independent algorithm. Figure 3 shows cumulative histograms of the best Random Forest and Boosting results in the dataset cross-comparison experiments. The Random Forest clearly outperforms the Boosting method when training on the XM2VTS dataset, although this is less obvious when training on BioID. Arguably, the large pose and illumination variation in the BioID dataset forces the Boosting algorithm to build more generic models, whereas overfitting occurs on the uniform XM2VTS dataset. Conversely, the Random Forest produces similar cumulative histograms in both cases, which may be seen as evidence for its resistance to overfitting.

(a) test set: BioID

				boost		forest			
training set		init	lin	pw	zs	vs	vs_1d	zs	zs_1d
BioID	median	7.3	6.5	**2.3**	2.5	2.5	2.4	4.0	2.6
	success	5.0	9.7	66.9	64.4	65.2	**71.5**	28.8	65.4
XM2VTS	median	7.3	16.7	7.2	5.5	3.5	3.3	5.2	**3.2**
	success	5.0	14.4	18.6	23.7	42.3	41.9	16.7	**45.3**

(b) test set: XM2VTS

				boost		forest			
training set		init	lin	pw	zs	vs	vs_1d	zs	zs_1d
BioID	median	7.8	16.0	2.3	2.6	2.2	**2.2**	4.9	2.2
	success	4.0	0.2	69.7	59.8	65.1	**74.3**	24.7	70.8
XM2VTS	median	7.8	1.8	1.8	2.0	**1.5**	1.6	4.3	1.7
	success	4.0	76.2	**87.2**	82.5	85.6	82.2	29.1	81.3

Table 2: Median of the displacement distribution and percentage of successfully converged samples using a full 5-stage **Independent** Sequential AAM model. The sequence was trained using datasets containing maximum displacements of $15, 10, 7, 3$ and 1.5 percent IOD, respectively. The two letter acronyms describe the different algorithms and are explained in the text. **Left:** Testing performed on the BioID database. **Right:** Testing performed on the XM2VTS database.

(a) test set: BioID

				boost		forest			
training set		init	lin	pw	zs	vs	vs_1d	zs	zs_1d
BioID	median	7.3	16.1	**2.4**	2.7	2.7	2.8	3.8	3.8
	success	5.0	0.1	**61.0**	57.7	59.0	54.8	35.3	35.3
XM2VTS	median	7.3	16.7	5.0	5.3	6.6	4.7	5.0	**4.0**
	success	5.0	14.4	29.5	22.4	22.1	27.6	19.4	**33.8**

(b) test set: XM2VTS

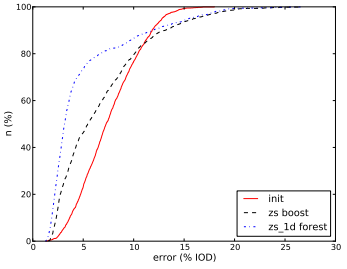
				boost		forest			
training set		init	lin	pw	zs	vs	vs_1d	zs	zs_1d
BioID	median	7.8	16.0	**2.6**	2.8	2.8	2.9	4.6	4.6
	success	4.0	0.2	**58.1**	54.9	55.3	52.8	30.4	30.4
XM2VTS	median	7.8	2.5	**1.6**	1.8	1.8	1.9	4.2	1.9
	success	4.0	69.0	**80.6**	78.8	77.5	70.2	34.6	71.6

Table 3: Median of the displacement distribution and percentage of successfully converged samples using a full 5-stage **Coupled** Sequential AAM model. The two letter acronyms describe the different algorithms and are explained in the text. **Left:** Testing performed on the BioID database. **Right:** Testing performed on the XM2VTS database.

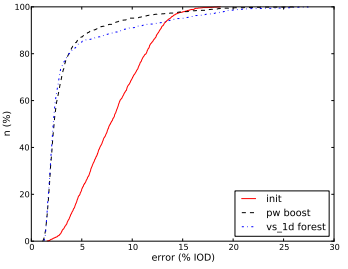
4.3 Timings

4.3.1 Training

The training procedures for both the Boosting and the Random Forest algorithms are computationally expensive. Even in our case, where only 1000 pixels were sampled from the



(a) train: XM2VTS, test: BioID



(b) train: BioID, test: XM2VTS

Figure 3: Cumulative error histograms of the best Boosting and Random Forest results when cross-comparing between the BioID and XM2VTS datasets.

images, more than 2 million Haar features need to be considered. In order to allow for reasonable processing times, we limited the amount of features to be considered to 50000 by random sampling. Thus the 5-stage Sequential AAM with the Boosting procedures required 6-8h to train. The Random Forest was implemented such that the trees were built in parallel using 12 processors on a server, resulting in training times ranging from 3-9h. All algorithms were implemented in C++.

4.3.2 Testing

	boost		forest			
	pw	zs	vs	vs_1d	zs	zs_1d
mean duration ($\pm$1 sd) (ms/iteration)	0.38 ± 0.02	0.41 ± 0.07	0.74 ± 0.04	1.39 ± 0.15	0.40 ± 0.07	1.42 ± 0.12

Table 4: Mean timings for one iteration of a Regression AAM comprising a simple pose-only shape model and the various regression algorithms.

In order to judge the efficiency of the algorithms involved, we timed the search for a single iteration in the pose prediction experiment using a release build of our C++ implementation on an Intel Core2 Quad desktop processor. The results are shown in table 4. As is expected, in particular the 1D implementations of the Random Forest are slower when compared to the Boosting procedures. However, these numbers were generated using a serial implementation of the Random Forest prediction. Since each tree in the forest is independent, the prediction could easily be parallelised, thus leading to an increase in performance proportional to the number of processors available.

5 Conclusion

In this paper we presented the use of Random Forest and Boosting regression algorithms within a Sequential AAM framework and discussed their use with two different feature selection criteria. We rigorously evaluated the algorithms by cross-comparing the performance of the resulting algorithms on two very different face datasets, XM2VTS and BioID. Though demonstrated for face analysis, AAMs are widely used in the medical domain, where we anticipate similar results will hold. We conclude that:

- the Random Forest shows superior generalisation performance
- vectorial data should be handled using independent 1D Random Forests
- independent training of Sequential AAMs is preferred over coupled training data

References

[1] BioID Database. URL https://support.bioid.com/downloads/facedb/index.php.

[2] XM2VTS Database. URL http://www.ee.surrey.ac.uk/CVSSP/xm2vtsdb/.

[3] Timo Ahonen, Abdenour Hadid, and Matti Pietik. Face Recognition with Local Binary Patterns. *ECCV 2004*, pages 469–481.

[4] Simon Baker and Iain Matthews. Lucas-Kanade 20 Years On: A Unifying Framework. *International Journal of Computer Vision*, 56(3):221–255, 2004.

[5] Aziz Umit Batur and Monson H Hayes. Adaptive active appearance models. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, 14(11):1707–21, 2005.

[6] L Breiman, J Friedman, C.J. Stone, and R.A. Olshen. *Classification and regression trees*. Chapman and Hall/CRC, 1984.

[7] Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32–32, 2001.

[8] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. BRIEF: binary robust independent elementary features. *ECCV 2010*, pages 778–792.

[9] T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001.

[10] David Cristinacce and Tim Cootes. Boosted regression active shape models. In *Proc. British Machine Vision Conference*, volume 2, pages 880–889, 2007.

[11] P. Dollár and, P. Welinder, and P. Perona. Cascaded pose regression. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010*, pages 1078 –1085, June 2010.

[12] N.D.H. Dowson and R. Bowden. Simultaneous Modeling and Tracking (SMAT) of Feature Sets. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, pages 99–105, 2001.

[13] J H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, 29(5):1189–1232, 2001.

[14] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression: a statistical view of boosting (With discussion and a rejoinder by the authors). *The Annals of Statistics*, 28(2):337–407, 2000.

[15] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, 2003.

[16] Hyung-Soo Lee and Daijin Kim. Tensor-based AAM with continuous variation estimation: application to variation-robust face recognition. *IEEE transactions on pattern analysis and machine intelligence*, 31(6):1102–16, 2009.

[17] Iain Matthews and Simon Baker. Active Appearance Models Revisited. *International Journal of Computer Vision*, 60(2):135–164, 2004.

[18] Jason Saragih and Roland Göcke. Learning AAM fitting through simulation. *Pattern Recognition*, 42(11):2628–2636, 2009.

[19] Jason Saragih and Roland Goecke. Learning active appearance models from image sequences. In *Proceedings of the HCSNet workshop on Use of vision in human-computer interaction - Volume 56*, VisHCI '06, pages 51–60, 2006.

[20] P. A. Tresadern, P. Sauer, and T. F. Cootes. Additive update predictors in active appearance models. In *British Machine Vision Conference*, 2010.

[21] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3:71–86, 1991.

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

[23] Shaohua Kevin Zhou and Dorin Comaniciu. Shape regression machine. In *Proceedings of the 20th international conference on Information processing in medical imaging*, IPMI'07, pages 13–25, Berlin, Heidelberg, 2007. Springer-Verlag.