Localizing Facial Keypoints with Global Descriptor Search, Neighbour Alignment and Locally Linear Models

Md. Kamrul Hasan[1], Christopher Pal[1] and Sharon Moalem[2]
[1]École Polytechnique de Montréal, Université de Montréal
[2]University of Toronto and Recognyz Systems Technologies
md-kamrul.hasan@polymtl.ca, christohper.pal@polymtl.ca, sharon@recognyz.com

Abstract

We present our technique for facial keypoint localization in the wild submitted to the 300-W challenge. Our approach begins with a nearest neighbour search using global descriptors. We then employ an alignment of local neighbours and dynamically fit a locally linear model to the global keypoint configurations of the returned neighbours. Neighbours are also used to define restricted areas of the input image in which we apply local discriminative classifiers. We then employ an energy function based minimization approach to combine local classifier predictions with the dynamically estimated joint keypoint configuration model. Our method is able place 68 keypoints on in the wild facial imagery with an average localization error of less than 10% of the inter-ocular distance for almost 50% of the challenge test examples. Our model therein increased the yield of low error images over the baseline AAM result provided by the challenge organizers by a factor of 2.2 for the 68 keypoint challenge. Our method improves the 51 keypoint baseline result by a factor of 1.95, yielding keypoints for more than 50% of the test examples with error of less than 10% of inter-ocular distance.

1. Introduction

The accurate localization of facial keypoints or landmarks has many potential applications. For example, the geometry of a face can be estimated by using these local points, which can be used to improve the quality of subsequent predictions for many different applications. For example, the face verification in the wild results posted on the well known Labelled Faces in the Wild (LFW) evaluation [11] confirm that essentially all top results require some form of face registration, and most of the top face registration techniques use facial keypoints. Another application in which accurate keypoints can dramatically improve performance is facial emotion recognition [19]. Recent work has also focused on emotion recognition in the wild [9].

Active Shape Models (ASMs) [5], Active Appearance Models (AAMs) [4], and Constrained Local Models (CLMs) [6, 15] involve the estimation of a parametric model for the spatial configuration of keypoints often referred to as shape models. AAMs typically use comparisons with images and image templates to capture appearance information in a way that can be combined with a shape model. In contrast, CLMs replace template comparisons with a per keypoint discriminative model, then search for joint configurations of keypoints that are compatible with a shape model. Older appearance based techniques have relied on only image features and have no explicit shape model. For example [21] takes a sliding window approach using Gabor features reminiscent of the well known Viola-Jones face detection technique and creates independent discriminative models for each keypoint. More recent work has used support vector regression for local appearance models and Markov random fields to encode information about spatial configurations of keypoints [20]. Other more recent work [22] has used a tree structured maximum margin Markov network to integrate both appearance and spatial configuration information.

Simple shape models can have difficulties capturing the full range of pose variation that is often present in 'in the wild' imagery. For this reason [22] uses a mixture of tree structured max margin networks to capture pose variation. They have also labelled a set of 205 images of 468 faces in the wild with 6 landmarks and released this data as the annotated faces in the wild (AFW) data set. Other work has dealt with the challenge of pose variation using a large non-parametric set of global models [1]. This work also released the Labelled Face Parts in the Wild (LFPW) data set. Other recent work by Dantone et al. [8] has quantized training data into a small set of poses and applied conditional regression forest models to detect keypoints. They have also labelled 9 keypoints on the LFW evaluation imagery and released the data for further evaluations.

There are a number of different performance measures

that have been used to evaluate the performance of techniques for keypoint localization. The L2 distance, normalized by the *inter-ocular distance*, is one of the most prominent metrics, being used in [1], [8, 20]. In terms of gauging the current state of the art performance [1], one of the most state of the art techniques [1] reports that 93% of the 17 keypoints of BioId [13] can be predicted with an average localization error of less than 10% of the inter-ocular distance, on the 29 points of the more challenging LFPW [1] only 90% can be predicted at the same 10% level. Dantone et al. [8] report that they are able to predict slightly below 90% of 9 keypoints they labeled for the LFW with error of less than 10% of the inter-ocular distance.

In contrast to *inter-ocular distance*, [22] uses a different relative error measure - the relative *face size distance*, which is actually the average of the height and width of a face detection window returned by a face detector. They have compared results with four popular contemporary models: the Oxford, Multi View Active Appearance Model (AAM), Constrained Local Models (CLMs), and a commercial system, from *face.com*. On the 68 point multi-PIE frontal imagery, they report that 100% of the keypoints can be localized with and error of less than 5% of the relative face size distance. For their Annotated Faces in the Wild (AFW) [22] dataset, only 77% of the 6 keypoints can be localized to the same level of error.

As two different relative error measures were used by [22] and [1], its difficult to compare their results. However, if we can co-relate these two measures: the *Inter-Ocular Distance* and the *Face Size*, it is possible to do a reasonable comparison. If we assume that the *Inter-Ocular Distance* is 1/3 the *Face Size*, then the results of [1] for the BioId dataset and the results of [22] for the MultiPIE dataset appear to be fairly close to one another. Although these results looks impressive, we have to remember that both BioId and MultiPIE are controlled databases with mostly frontal images. If we use the same heuristic conversion, we see that [1] appears to be able to do a better job than [22] for real world datasets, however we must compare across the LFPW and AFW data sets as well leading to too much uncertainty to really gauge the relative performance of these techniques. For these reasons there is a clear need to organize a controlled challenge comparing keypoint detection techniques using in the wild imagery. The 300-W challenge has thus been organized to address this need and we now present the approach we have taken in our submission to the challenge.

2. Our Approach

We first summarize our overall approach then discuss the algorithmic details of the different parts of the overall system.

[1] prior to the 300-W challenge evaluation

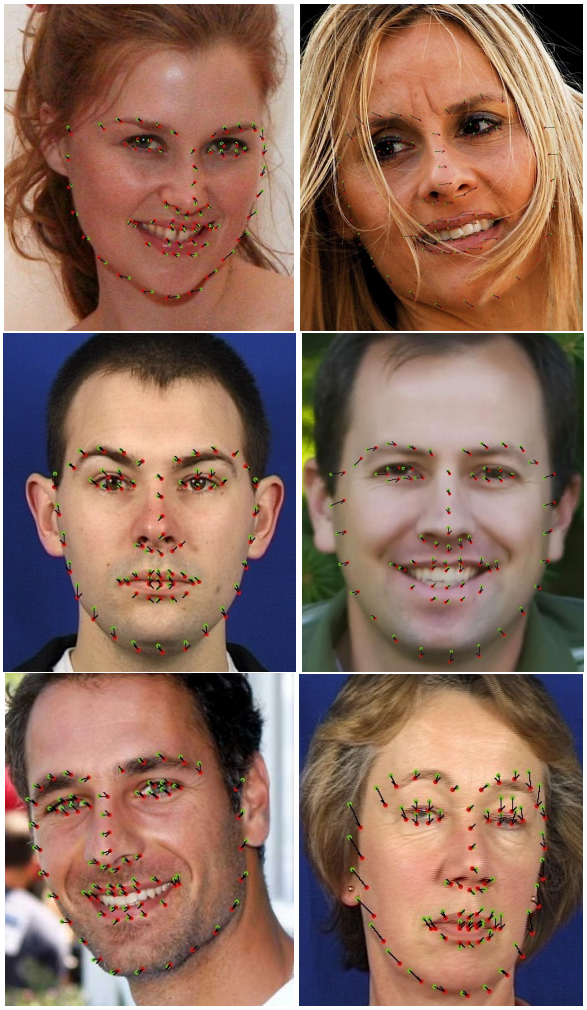

Figure 1. Green coloured keypoints are produced by a nearest neighbour model, while the red coloured keypoints are generated through our model. Arrows from green points, connecting the red points, show the keypoints movement directions during optimization by our model.

2.1. A High Level View of Our Approach

Using the competition training set we first train a set of local discriminative SVM based classifiers for each keypoint using fine scale histogram of oriented gradient (HoG) based descriptors. We then create a database of all training images using a coarser scale or more global HoG descriptor for each face based on the provided bounding box location. For a given test image, using the global HoG descriptors we find the $N = 100$ nearest neighbours from within the training set database. We project the global HoG descriptor down to $g = 200$ dimensions for this task. Using the top $M = 3$ closest neighbours in the database we compute the average location of their corresponding labelled keypoints and use these locations to restrict the application of each keypoint detector to a small local window of size $n \times m$,

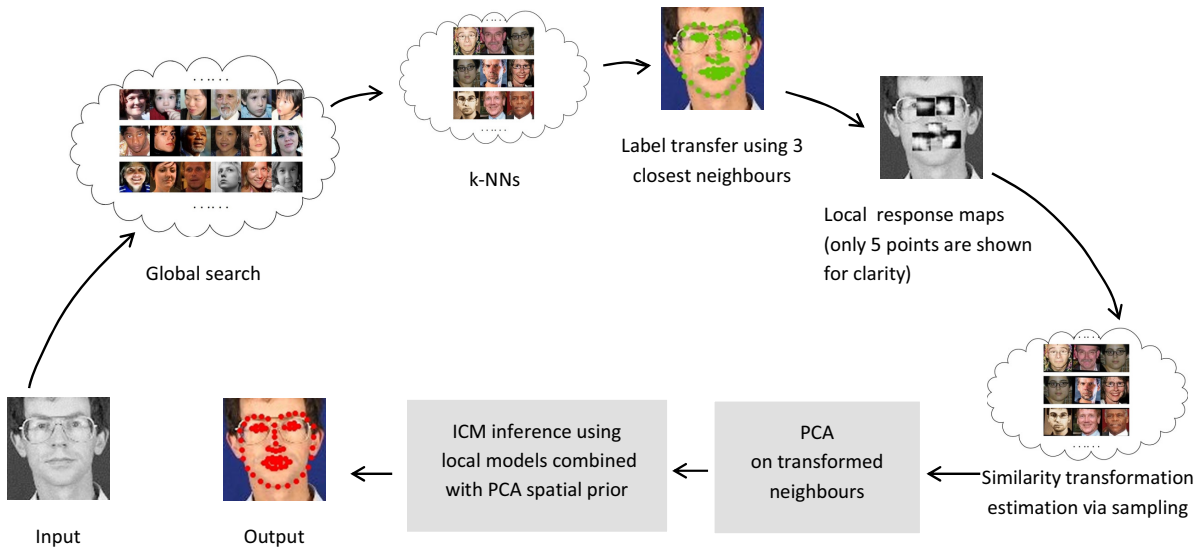

Figure 2. Our complete pipeline.

with $n = m = 16$. This procedure yield a set of response images for each keypoint. We identify $k = 2$ modes per response image using a non-maximal suppression technique. Using the modes identified for each keypoint we then use a Random Sample Consensus (RANSAC) like method to estimate similarity transforms for each of the 100 nearest neighbours. Using these 100 neighbours (registered to the input face via similarity transforms) we perform a probabilistic principal component analysis (PPCA) and keep $p = 30$ dimensions. We then initialize a iterated conditional modes (ICM) based search procedure using the mean of the top $t = 10$ best aligned exemplars as our starting point. This ICM based search is thus performed using the scores provided by the set of 68 response images for each keypoint using the dynamically estimated PPCA model to encode spatial interactions or the shape model.

2.2. Initial Nearest Neighbor Search

We wish to accelerate the search for each keypoint based on a local classifier as well as accelerate a more detailed inference procedure that combines local keypoint predictions with a separate model for valid global configurations of keypoints estimated from nearby exemplars. Our intuition and hypothesis here is that if we have a large database of wild faces with keypoint labels (covering many identities, poses, lighting conditions, expression variations, etc.), a simple nearest neighbours search using an effective global descriptor should be able to yield exemplars with keypoint locations that are also spatially close to the correct keypoint locations for a query image. In figure 3, for each query image on the left, we show the 3 nearest neighbours, followed by the mean of their corresponding keypoints on the right. We can clearly see that the level of pose and expression correspondence between the query and returned results is reasonable. From this analysis one can see that this approach appears promising.

Given an initial estimate of keypoint locations, we can dramatically reduce the amount of time needed to execute local per-keypoint classifiers by restricting their search to a small window of plausible locations. Further, we can determine an appropriate size for such a window via cross validation techniques. Additionally, while this nearest neighbour technique might not be able to provide an exact solution to the keypoint placement problem, neighbours returned by this technique could be brought closer to the correct solution through estimating a simple (ex. similarity) transform. For this step we use candidate keypoint locations obtained from local classifiers and use a RANSAC like method reminiscent to [1]. However, here this estimation can be done with far greater efficiency since here we shall only consider a small set of $N = 100$ neighbours as opposed to the use of a random sampling strategy over the entire data set. Finally, once we have this set of spatially registered neighbours we can then build a more accurate model of the spatial distributions of their keypoints. This initial nearest neighbour step itself could indeed yield an initial solution to our keypoint placement problem and we shall provide some comparisons with this type of approach as a baseline comparison.

To build both our global descriptors and our local classifiers we need image features. We have found that Histograms of Oriented Gradients or HOG features [7] are extremely effective for face pose detection and identify recognition. As one of the goals of our first filtering step is to filter away dissimilar (in pose, expression, and identity) exemplars, we used HOG features for our global descriptor. In particular, we extracted HOG features from overlapping

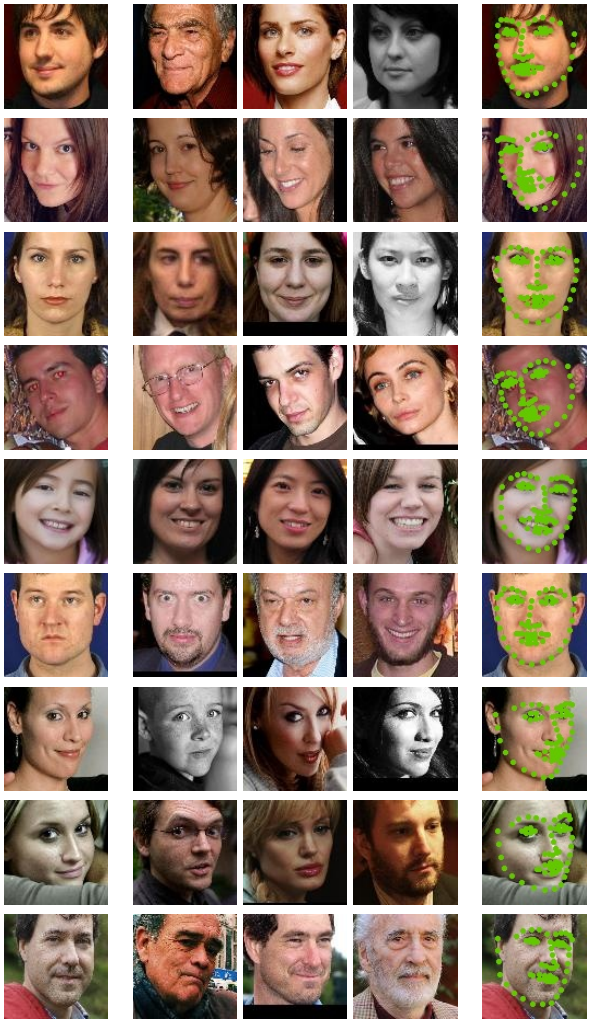

Figure 3. Query faces (first column), corresponding three nearest neighbours (columns: 2-4), and label transfer results by simple averaging (column 5).

patches on the image grid, and concatenated them to generate a global feature for a face. The grid intervals and the patch size were determined through a grid search and cross-validation. We compared the closeness of keypoints for images returned via this approach to input queries, varying the HOG block size, and the amount of overlap. As a result of this procedure for our subsequent experiments we used a block size of 24x24, and the intervals between blocks was 12. We also used a Principal Component Analysis (PCA) projection for HOG features, and reduced the dimensionality to 200. There were just over 6,000 training faces in the 300-W evaluation [12]. Brute force search using 200 dimensional vectors was therefore quite feasible, taking less than a second.

As discussed, we would like to both transfer labels from these returned results to provide a baseline technique as well as use the transferred labels to restrict our local search using keypoint classifiers. We could choose the first match or aggregate results from first M matches. Using cross validation we found that an average from the first 3 matches gave us the best label transfer results. In our subsequent experiments, we therefore aggregate results from the top 3 matches.

2.3. Defining Search Regions for Local Classifiers

As outlined above, we use binary SVMs with HOG features as input to our local classifiers. Classifiers are only applied within an input window defined by averaging of keypoint locations for the top $M = 3$ neighbours returned by the nearest neighbour search using a global image descriptor. We used features that were extracted from image patches of size 24x24. For each keypoint, 2000 positive patches, centred at each keypoint location and 2000 negative patches from elsewhere in the image were used as our training data. Half of the negative patches were selected from closer locations; more specifically, these 50% negative patches were selected by selecting the patch centre falling within the 7x7, but not 5x5 region around the keypoint. The other 50% were selected from other random locations. See the corresponding step in the pipeline of Figure 2 to see the relative size and locations of these windows.

2.4. Fast Registration of Neighbours

We wish to improve the alignmnets of the keypoints on an input query image and the keypoints on our set of nearest neighbours returned via global descriptor search. We shall use these exemplars after this (2D similarity) alignment later to produce a more accurate statistical model of empirically plausible distortions from the mean of these 100 keypoint sets. However, we of course do not yet have correct keypoints for our query. We do however have candidate locations that can be extracted from the response images associated with the spatially restricted search using keypoint classifiers. We use a separate Support Vector Machine (SVM) per keypoint to produce these local response images, $\{d^i\}_i^n$. As in [1] and other RANSAC based techniques, we then randomly select two points from a random exemplar image found with our nearest neighbours, then perform a similarity warp using the two corresponding modes from the response images.

A similarity transformation has three parameters, $\{$translation, scaling, and rotation$\}$. As the human face is a 3D object, the true face mesh defined through the fiducial points on it is also a 3D object, it is therefore difficult for a 2D similarity transformation to align a pair of 2D facial images with one another if they are from different poses. However, as discussed above and as seen in figure 3 our nearest neighbour method is able to to filter away faces from dramatically different poses and thus reduces our search space extensively. This 2D similarity registration step thus ac-

counts for minor differences in images that can addressed by simple 2D rotations, scale changes, translations and reflections. The intuition is that we would like to account for these effects prior to capturing other more complex factors of variation using our locally linear (PCA) modeling technique discussed in section 2.5. Our search algorithm is provided below.

Exemplar Warping and Search Algorithm

1. For a test face, generate n response images, $\{d^i\}_{i=1}^n$, for n keypoints using corresponding local classifiers.
2. Extract two modes per response image using a non-maximal suppression technique to create a putative list of keypoints.
3. From the putative keypoint list, select a random pair and :
   - Take a random exemplar from the 100 nearest neighbours provided by the methodology, described in section 2.2.
   - Estimate a similarity transform to align the test face with the exemplar using two random modes from two random response images and the corresponding exemplar keypoints.
   - Evaluate the point distortions between these two images using the L2 distance function, $d_{k,t} = (X^l - X^l_{k,t})^T(X^l - X^l_{k,t})$; where, X^l is the vector of n response image maximum modes, and $X^l_{k,t}$ is the corresponding warped locations of exemplar $X_{k,t}$ on X^l.
4. Iterate step 3, $r = 10,000$ times and store the results.
5. Select the best fit N exemplars, $\{X_{k,t}\}_n^N$. Transform all their corresponding keypoints using transformation parameter, t. This results N warped exemplars on the test image for our next step.

2.5. Combining Local Scores with a Spatial Model

To combine the uncertain predictions for each model with a model of likely spatial configurations we first estimate a locally linear PCA model dynamically using these N warped exemplars.

To be more precise, for a given image I, where $\mathbf{x} = [x_1, x_2, \cdots x_n]^T$ and $\mathbf{y} = [y_1, y_2, \cdots y_n]^T$ give the x and y co-ordinates of each of the n keypoint locations, we wish to combine the output of a local classifier with a spatial model for global keypoint configurations. Let, $D = \{d^1, d^2 \cdots, d^n\}$ be the response images, generated by these local classifiers. A response image, d^i, defined here is simply a 2D array with binary prediction probability for any pixel in the test image being classified as the correct location for point i by the i^{th} local classifier. For a visualization of the response image probability values see corresponding step of Figure 2 (local response maps) where probabilities are scaled by a factor of 255 (8 bit gray-scale images). Let the log of the score for the positive prediction for keypoint p at the corresponding gridpoint location g_x, g_y be defined as $s_p(g_x, g_y)$.

We use a probabilistic variant of PCA and correspondingly use the log of a Gaussian distribution with a factorized covariance matrix to couple local prediction via the spatial interaction terms of an energy function with the following form:

$$E(\mathbf{x}, \mathbf{y}) = \tag{1}$$
$$-\sum_{p=1}^{N}\sum_{g_x=1}^{n}\sum_{g_y=1}^{m} s_p(g_x, g_y)\delta(x_p - x'_p(g_x))\delta(y_p - y'_p(g_y))$$
$$+\frac{1}{2}([\mathbf{x}^T\mathbf{y}^T] - \mu^T)(\mathbf{W}\mathbf{W}^T + \mathbf{\Sigma})^{-1}([\mathbf{x}^T\mathbf{y}^T]^T - \mu),$$

where $\mathbf{W}$ corresponds to the eigen vectors of the PCA, $\mathbf{\Sigma}$ is a diagonal matrix, μ is simply the mean of the $N = 100$ nearest neighbours returned from the global descriptor search after RANSAC similarity registration, $x'_p(g_x)$ and $y'_p(g_y)$ are the x and y locations for keypoint p corresponding to grid indices g_x and g_y. To minimize E we must perform a search over the joint configuration space defined by each of the local grids of possible values, $x_p \in x'_p(g_x), y_p \in y'_p(g_y)$ for each keypoint p.

While we have formulated our approach here as an energy function minimization technique, one might equally formulate an equivalent probabilistic model encoding spatial configurations of keypoints as a real valued distribution, with intermediate variables that transform the model into into discretized grids, followed by a final conversion into binary variables for each position on the equivalent grid. One could then use the SVM scores as a form of soft evidence concerning these binary variables.

2.6. Inference with the Combined Model

We used an Iterative Conditional Modes (ICM) based minimization procedure to optimize Equation (1). Starting with an initial assignment to all keypoint locations, we iteratively update each keypoint location x_p, y_p.

Fitting algorithm :

1. Take the average of the keypoint locations for the N aligned neighbours and initialize the initial solution as X^*.
2. Iterate until none of the keypoints in $\mathbf{x}$ and $\mathbf{y}$ moves or a maximum number of iterations is completed (we used c=10):

(a) Select a keypoint, (x_p, y_p) from X^*.

(b) Minimize Equation (1) using $x_p^*, y_p^* = \arg\min_{x_p, y_p} E(\mathbf{x}, \mathbf{y})$.

(c) Update, $x_p \leftarrow x_p^*$, and $y_p \leftarrow y_p^*$.

3. Take X^* as the output.

2.7. Data Pre-processing

We used both the true and the predicted bounding box initializations for all five datasets : AFW [22], helen [14], ibug [12], lfpw [1], XM2VTS [17], provided as the 300-W challenge benchmarks [18, 12]. By predicted face bounding box, we mean the face bounding boxes predicted by an i-Bug [12] version of the face detector [22]. We cropped faces from these images with additional 20% background for each side. The cropped faces were then re-scaled to a constant size of 96x96 resolution using bilinear interpolation.

3. Experiments and Results

3.1. Cross Validation Experiments

Before discussing the challenge results produced by our algorithm, we first provide here our five fold cross-validation results for the following three configurations:

- Using 300-W true-true (tt) face bounding boxes.
- Using 300-W true-predicted (tp) face bounding boxes.
- Using 300-W predicted-predicted (pp) face bounding boxes.

These experiments allow us to see the impact of errors in bounding box predictions. The left and right panesl of Figure 4 show these five fold cross validation results for the two 300-W labeling tasks: (a) the 68 point labeling problem, and (b) the 51 points labelling problem. For each cross-validation step, we only kept the XM2VTS database faces in the validation set, but removed from the training set. This particular configuration was followed due to an observation that XM2VTS is the only non wild database out of the five 300-W evaluation databases.

300-W true-true (tt) face bounding boxes

For this particular setup, all models were cross-validated (five fold) using the true face bounding box initializations. In other words, this setup assumes that the provided face detector is always perfect in detecting faces. As a result, this configuration might be thought as an upper bound of our model when it deals with faces provided by a face detector with some level of face detection noises.

Our simple nearest neighbour label transfer resulted 79% of the keypoints having error less than 10% of relative inter-ocular distance for the 68 points problem. This relative error was 81% for the 51 keypoints problem. For the same setup our complete method predicted 85% and 90% of the keypoints within a 10% relative inter-ocular distance error margin.

300-W true-predicted (tp) bounding boxes

For this setup, while cross-validating, the models were trained using true face bounding box initializations, and validated using face bounding boxes that were provided by the 300-W challenge face detector. For the 68 keypoints problem, our nearest neighbour model predicted 70% of the keypoints within the 10% inter-ocular error margin. This relative rate was 75% for the 51 keypoints labelling task. We can clearly see the drop of performance, compared to the upper bound (tt) setup. Interestingly, our model was able to achieve levels of precision, very close to the upper bounds. These rates were 83% and 89% respectively for the 68 and 51 keypoint problems. Analyzing Figures 4 left and right it appears that our method is fairly robust to face bounding initialization errors that might arise as a result of a noisy face detector.

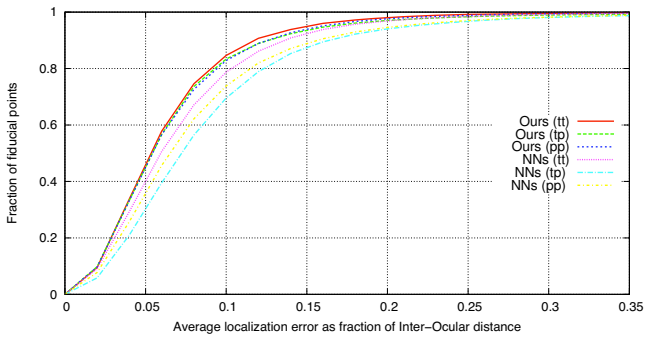

Figure 4. Average localization error as fraction of inter-ocular distance for 68 fiducial points

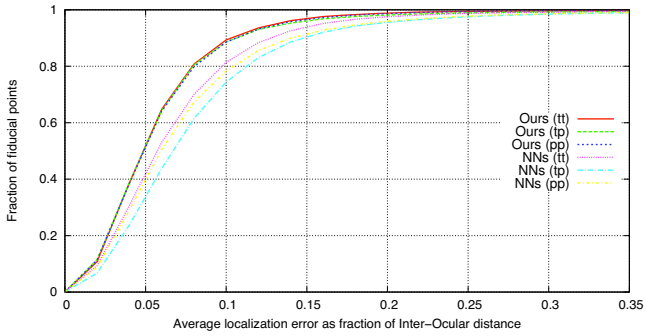

Figure 5. Average localization error as fraction of inter-ocular distance for 51 fiducial points

300-W predicted-predicted (pp) bounding boxes

For this particular setup, all models were cross-validated (five fold) through the predicted face bounding boxes, provided by the 300-W challenge face detector. Here again

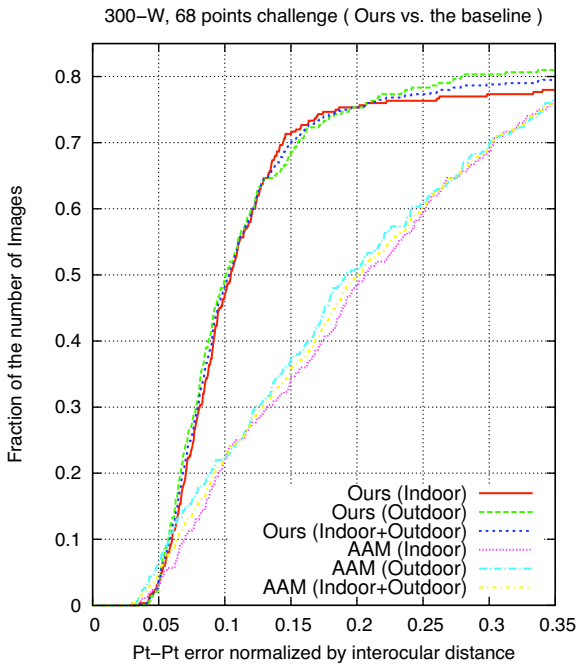

Figure 6. Average localization error as fraction of inter-ocular distance for 68 fiducial points

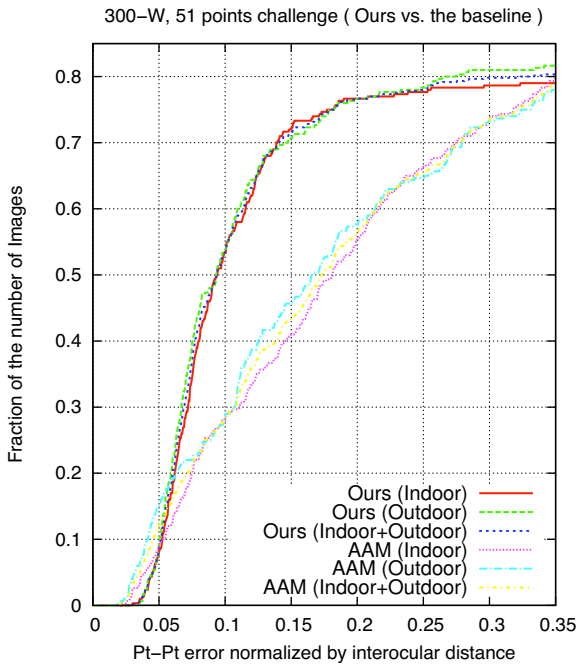

Figure 7. Average localization error as fraction of inter-ocular distance for 51 fiducial points

results are compiled in the left and right panels of Figure 4 for the 68 and 51 keypoint problems respectively. Our nearest neighbour model did better for this setup, compared to setup two (tp). For the 68 keypoints problem, it was able to produce 74% of keypoints within the 10% inter-ocular margin. This rate was 74% for the 51 keypoints labelling task. Our complete method achieved similar levels of precision as of setup two (tp) for both 68 and 51 keypoints labelling problems. It seems, our complete approach is less sensitive to face detection errors.

3.2. 300-W challenge results

Our model that was submitted to the 300-W challenge was trained using all the images from the 300-W evaluation, except the XMT2VTS database faces. The 300-W challenge organizers ran our algorithm on a unreleased dataset of 300 indoor and 300 outdoor images, and returned the results produced by our algorithm. They also provided their baseline results for the same test data. In their evaluation they have used a slightly different metric than our earlier evaluation. More specifically, the challenge used a per image, rather than a per point relative error evaluation. The same inter-ocular distance is used as a relative metric; however, the point to point distances for all points are averaged for an image and performances are reported on % of the number of images.

The baseline provided by the 300-W challengeis an implementation of the Inverse Compositional Active Appearance Model (AAM) algorithm [16] using the edge-structure features of [3].

Figures 6 and 7 were generated using the returned results. For the 68 points protocol and for both the indoor and outdoor settings, the Inverse Compositional AAM baseline was able to produce all keypoints for 22% of the test images with an average error of less than 10% relative inter-ocular distance. While testing with images from indoor environments, for the same 68 points protocol and same relative error, our algorithm was able to produce correct results for 47% of the images. For outdoor images, this percentage even improved to 50% for the same protocol and relative error metric.

For the 51 points problem, for a 10% relative inter-ocular relative error metric, our algorithm was able to produce all keypoints for 52% indoor test images. This rate was 53% for outdoor images for the same set-up. In comparison, the baseline AAM was only able to produce all keypoints for 27% images for both indoor and outdoor settings.

For both the 68 and 51 keypoint problems, for a combined indoor and outdoor settings, our model was able to produce all keypoints for over 72% of the images for an average relative inter-ocular distance of 15%. This rate was 35% (for the 68 keypoint problem) and 43% (for 51 keypoints) respectively for the same relative measure using the baseline AAM. This means, for about 3, out of 4 real-world images, our algorithm can produce all fiducial points within an average of 15% inter-ocular. For the same setting, the Inverse Compositional Active Appearance Model (AAM) can do it for less than 2 images.

3.3. Runtime analysis

On an x86_64, 3.26 GHz machine with 15.67 GB RAM, our pipeline takes about just over a minute to produce the 68 keypoints as outputs. We obtained the following average runtimes for each of the key steps in our pipeline: (a) Global search + k-NNs + label transfer: **0.5** seconds, (b) Local response images: **32** seconds, (c) Similarity transformation estimation via sampling: **36** seconds, and (d) PCA + ICM : **6** seconds. We can see that the steps (b), the computation of local response images, and (c), the similarity transformation estimation via sampling take over 90% of the runtime. Both the computation of local response images and the similarity estimation steps are operations that could be parallelized using popular Graphical Processing Unit (GPU) acceleration techinques or multi-core methods. Accelertions by an order of magnitude are often obtainable through these techniques and we therefore believe the run time for this method could be reduced from over a minute to a couple of seconds.

4. Discussion and Conclusions

In this research, we have outlined our approach used to submit results to the 300-W facial keypoints localization challenge. Our method is able to place 68 keypoints on in the wild facial imagery with an average localization error of less than 10% of the inter-ocular distance for almost almost 50% of the test examples and about 70% of the images with a average localization error of less than 15% of the inter-ocular distance. In comparison the AAM baseline provided by the challenge yielded only 23% and 35% of the images with these levels of error. On the 51 keypoint problem our method yields 53% and 73% of the images at the 10% and 15% levels of error, which also compares favourably to 29% and 34% yielded by the challenge baseline AAM.

5. Acknowledgements

We would like to thank the people releasing the tools in [2], and [10] which we used in this project. We would also like to thank the 300-W challenge organizers, (and especially Dr. Georgios Tzimiropoulos), for their guidance throughout the competition. We thank NSERC for funding under the Engage program.

References

[1] P. Belhumeur, D. Jacobs, D. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. In *CVPR*, pages 545–552, 2011. 1, 2, 3, 4, 6

[2] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available: http://www.csie.ntu.edu.tw/~cjlin/libsvm. 8

[3] T. Cootes and C. Taylor. On representing edge structure for model matching. In *CVPR(1)*, pages I–1114–1119, 2001. 7

[4] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In *ECCV*, 1998. 1

[5] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models-their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995. 1

[6] D. Cristinacce and T. Cootes. Feature detection and tracking with constrained local models. In *BMVC*, pages 929–938, 2006. 1

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR (1)*, pages 886–893, 2005. 3

[8] M. Dantone, J. Gall, G. Fanelli, and L. V. Gool. Real-time facial feature detection using conditional regression forests. In *CVPR*, 2012. 1, 2

[9] A. Dhall, R. Goecke, J. Joshi, M. Wagner, and T. Gedeon. Emotion recognition in the wild challenge 2013. In *ICMI*, 2013. 1

[10] L. He, (2011), Histograms of Oriented Gradients Feature Extraction. Available: http://mloss.org/software/view/346/. 8

[11] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, UMASS, Amherst, October 2007. 1

[12] iBUG, (August, 2013). Available: http://ibug.doc.ic.ac.uk/resources/300-W/. 4, 6

[13] O. Jesorsky, K. J. Kirchberg, and R. Frischholz. Robust face detection using the hausdorff distance. In *AVBPA*, pages 90–95, 2001. 2

[14] V. Le, J. Brandt, Z. Lin, L. D. Bourdev, and T. S. Huang. Interactive facial feature localization. In *ECCV (3)*, pages 679–692, 2012. 6

[15] S. Lucey, Y. Wang, M. Cox, S. Sridharan, and J. Cohn. Efficient constrained local model fitting for non-rigid face alignment. *Image and vision computing*, 27(12):1804–1813, 2009. 1

[16] I. Matthews and S. Baker. Active appearance models revisited. *IJCV*, 60(2):135–164, 2004. 7

[17] K. Messer, J. Matas, J. Kittler, J. Lüttin, and G. Maitre. Xm2vtsdb: The extended m2vts database. In *AVBPA*, pages 72–77, 1999. 6

[18] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. A semi-automatic methodology for facial landmark annotation. In *CVPR Workshop*, pages 896–903, 2013. 6

[19] Y.-I. Tian, T. Kanade, and J. F. Cohn. Recognizing action units for facial expression analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(2):97–115, 2001. 1

[20] M. Valstar, B. Martinez, X. Binefa, and M. Pantic. Facial point detection using boosted regression and graph models. In *CVPR*, pages 2729–2736, 2010. 1, 2

[21] D. Vukadinovic and M. Pantic. Fully automatic facial feature point detection using gabor feature based boosted classifiers. In *SMC*, pages 1692–1698, 2005. 1

[22] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *CVPR*, pages 2879–2886, 2012. 1, 2, 6