A Scale Robust Calibration Method for Face Detection Framework

Pedro H. R. Assis
Departamento de Informática
PUC-Rio
Rio de Janeiro - RJ, Brazil
Email: passis@inf.puc-rio.br

Mário F. M. Campos
Departamento de Ciência da Computação
Universidade Federal de Minas Gerais
Belo Horizonte - MG, Brazil
Email: mario@dcc.ufmg.br

***Abstract*—This paper describes an algorithm for finding a scale that should be used to improve the performance of face detection framework if image resizing would be applied as a pre processing step. The algorithm is based on a supervised machine learning approach. It iteratively resizes the image until it reaches the scale which most closely satisfies certain desirable conditions. The impact of interpolation kernels on finding a scale was also empirically analyzed. A set of experiments for the calibration method proposed using a face detection framework is presented. We also showed in our experiments that our method reduced the occurence of false positive detections in comparison to the usual non-resizing detections.**

***Keywords*-face detection; optimization; object-detection framework; computer vision**

I. Introduction

Object detection is one of the most studied problems in Computer Vision. Some solutions are widely used in several number of applications ([1],[2]). A large number of them deals with a vast amount of data, e.g. social networks featuring users image processing, or simply applications that perform real-time detections on large images, e.g. security systems, augmented reality applications. Therefore, any performance improvement on the detection time drastically affects the development of such applications.

Also, there is an increasing evolution in camera resolution which implies that there is an increase of object detection time. This justifies adjustments on the actual object detection framework to follow the growth of the resolution of input images.

One of the shortcomings of the validation of this work was the use of a well behaved data set, where images did not differ much from each other. The results for this data set were more expressive than for a data set of a heterogeneous image which showed a better performance for small groups of objects to be detected.

Contributions: We explored the following problem: *"Given a set of images of same resolution and a object detection framework, to find a stable scale such that applying a pre and post-scaling in all the images, it reduces the total detection time with no loss of robustness."*

We present a calibration method that attempts to solve the problem stated above. We also compared the influence of the performance and of the robustness of four different interpolation kernels in the resizing steps. In our experiments, we evaluate gains of robustness and performance of our method using a face detection framework in two different data sets.

A. Related work

The method presented in this work may be applicable to several object detection approaches which may be affected by the image resolution. For this work we chose a face detection framework. In this section we discuss several strategies already proposed to improve the performance of face detection and face recognition systems. Analogously, their theory is also applied to general object detection framework.

In [3], a new interpolation kernel is proposed: the so-called Decimation Algorithm. The authors propose a method for finding a unique scale reduction value for every image in a dataset. A Gaussian pyramid [4] is used to find an optimal resolution which achieves the best performance. Every image in the dataset is associated with its best resolution scale. Our approach, on the other hand, calculates the best resize scale using only one image in the set that is used for all the images of the set.

An opposite approach for face recognition optimization is proposed in [5]. In this case, a solution was proposed for the problem of adapting a system designed for low resolution images to accept high resolution images as input. This approach is very effective for high resolution videos and their results are a success case of gains of performance and robustness by varying the input image dimensions.

In [6], an analysis of the effects of image resolution on the performance of face recognition algorithms is presented. It is concluded that the image dimension is inversely proportional to the performance of the object detection framework. In order to find the optimal scale it was proposed a feature descriptor that is scale-robust in which could be a possible approach. Algebraically, the problem is reduced to a minimization problem and it requires modifications in the object detection framework. In our solution we treat the object detection framework as a "black-box." Our approach only performs a pre and post processing in images.

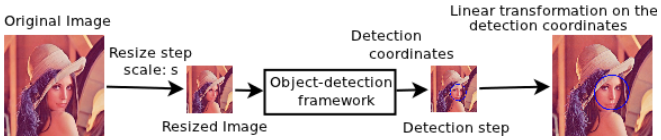

Fig. 1. The overview of the proposed technique. The resized image is processed by the framework and the coordinates of the faces detected are mapped to the original image.

B. Technique overview

The input image is resized by a scale s and the detection occurs on the resized image. The detections coordinates given as output are inputs of a linear transformation that maps the coordinates back to the original image. This scheme is illustrated in Fig. 1.

The scale s has to satisfy a gain in performance and guarantee no loss of robustness. Its calculation occurs only once for a selected image from a set of images. The selected image is such that the detection is more sensitive to scale variations. We call this the "worst-case" image of the set. The scale s is calculated by an iterative algorithm which converges under certain assumptions over the object detection framework properties.

II. Technical Background

In this section, we detail some definitions that we use to build our algorithm to find the best scale s.

Let ϕ be a function $\phi : \mathbb{R}_+ \to \mathbb{N}$. It is an immediate conclusion that:

Lemma 1. *Let ϕ be a function such that $\phi : \mathbb{R}_+ \to \mathbb{N}$ then ϕ is a non-differentiable function.*

In order to explore the graphical properties of ϕ we need the following definition:

Definition 1. *(Step function) A function $f : D \to C$ is called a step function if it can be written as:*

$$f(x) = \sum_{i=0}^{n} \alpha_i \chi_{A_i}(x), \forall x \in D.$$

where $n \geq 0$, $\alpha_i \in D$, A_i are intervals, and χ_A is the indicator function of A:

$$\chi_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases}$$

Now, we want to define the domain and the counter-domain of ϕ in a given positive interval. Let $\mathbb{R}_+(s)$ be the closed real interval $[0, s]$ and $\mathbb{N}(d)$ the closed natural interval $[0, d]$. Finally, we claim that ϕ is a decreasing function. So we have the following:

Lemma 2. *Let ϕ be a decreasing function such that $\phi : \mathbb{R}_+(s) \to \mathbb{N}(d)$ and $s \neq \infty$, $d \neq \infty$ then ϕ is a step function.*

Proof: As $\mathbb{N}(d)$ is a finite discrete set and ϕ is a decreasing function, we have $\phi(s_1) \leq \phi(s_2), s_1 < s_2$. Therefore, we can define the limited intervals $A_i = [a_{i-1}, a_i[$, such that $a_{i-1} \leq a_i$ and $\phi(a_i) = i + 1$. Hence, we can write ϕ as:

$$\phi(x) = 1\chi_{A_1}(x) + 2\chi_{A_2}(x) + \cdots + d\chi_{A_d}(x) \qquad \square$$

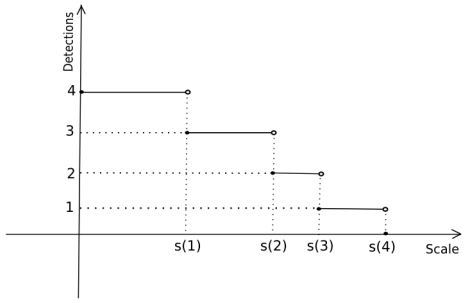

Fig. 2. An example of a ϕ function illustrating its decreasing step function form. The x-axis represents the scale factor which is used to reduce the original image. The y-axis represents the number of detections provided by the given detection framework.

Informally, there are more elements in $\mathbb{R}_+(s)$ than in $\mathbb{N}(d)$, so if ϕ is a decreasing function, by the pigeonhole principle $\exists x, y \in \mathbb{R}_+(s), x \neq y$, such that $\phi(x) = \phi(y)$. Hence as the counter-domain is discrete and ϕ is a decreasing function, ϕ must be a step function. An example of a function ϕ is sketched in the Fig 2.

A. Training Images

In order to state our algorithm in the next section, we define the concept of training image.

Definition 2. *(Training image) Let I be a set of images of same dimension, and d_i be the greatest distance between an object to be detected in the scene and the camera that produces $i \in I$. So i is a training image $\bar{i}$ if $\{\forall j \in I - \{\bar{i}\} \mid d_j < d_{\bar{i}}\}$.*

Informally, a training image is the "worst-case" to be detected in the set of images I. In general, $\bar{i}$ contains the smallest object (in dimension) to be detected in comparison to all the objects present in the other images of I. Our experiments show that if I is homogeneous in d_i, i.e. $|d_i - d_j| < \theta$, for a small θ, a random image can be chosen as a training image.

III. New Technique

In this section we present our calibration method.

A. Formulation

Initially, we define one necessary property that the object detection framework must satisfy:

- *Decreasing property:* the number of object detections must be inversely proportional to the image dimension.

This property is very common since most kind of features extracted from the image are scale variant. For example, we can shrink an image enough that an object becomes no longer detected.

The key is to find an image scale such that there is no decrease of robustness, informally, the image is not reduced enough to vanish an object. In order to solve this problem,

the relation between the number of detections and the image scale has to be explored.

For a given input image, object detection framework, by definition, always return a unique natural number of detections. Hence we can infer that the relation between the number of detections and image scales is a function. Considering that the framework satisfies the decreasing property, we claim that the function that maps the scale of a image resolution to its number of object detections is a decreasing step function $\phi : \mathbb{R}_+(s) \to \mathbb{N}(d)$ as defined in Section 2.

B. Solution

Our algorithm explores iteratively the ϕ function. We look at the ϕ function generated by the training image $\bar{i}$ as defined in Section 2. At this point, a supervised step is required, and the number of objects detected D, excluding false positives from $\bar{i}$ must be provided. According to the conventions stated in Section 2, we are interested in:

$$\phi^{-1}(D) = A_D = [a_{D+1}, a_D[$$

which our algorithms uses to return the best scale s as:

$$s = \frac{a_{D+1} + a_D}{2}$$

We consider the best scale the mean value of the interval represented by $\phi^{-1}(D)$ for stability matters. Any other value on the extremes are susceptible to generate false positives or to vanish one of the correct detections. Therefore, s is the scale rate used to resize all the images in I, i.e. length and width are divided by s.

IV. Implementation

Algorithmically, we calculate the limits of A_D using an iterative approach. Basically, we start by $s = 1$, increasing it at each iteration by a constant ϵ. At every iteration, we call the object detection framework and compare the number of detections with the desirable value D. We save the first and the last occurence of D, and then we take the average.

As a trade-off, if ϵ is too small, the convergence will take more time, but the precision will be increased. If ϵ is too large, the convergence will be quick but the precision may not be granted.

Formally, the pseudo-code 1 states our calibration method.

Algorithm 1: Calibration method

input : A training image $\bar{i}$, the number of objects to be detected D and an iterative step constant ϵ.
output: A scale rate

```
s ← 1
left ← 1
right ← 1
left_defined = false
d = number_of_detections(ī)
while d ≥ D do
    if d = D and left_defined = false then
        left ← s
        left_defined = true
    if left_defined = true then
        right ← s
    s ← s + ε
    ī ← scale(ī, s)
    d = number_of_detections(ī)
return (left + right)/2
```

V. Experiments

A. Image datasets

For experiments, we used two datasets. First, we applied our method on the Caltech frontal face dataset [7]. This database was collected in 1999. It contains 450 frontal face images of 896×592 pixels in a JPEG format. There are pictures of 27 individuals under different lighting, expressions and backgrounds.

The Caltech dataset was interesting for our experiments because all the faces are not cropped and the background of the images contribute to a considerable number of false positive detections. Some examples of images from the dataset are in Fig 3. We verified improvements proposed by our theory using this dataset.

Fig. 3. 20 of 592 images of the Caltech Dataset

The second dataset we used was The Images Groups Dataset [8] provided by Cornell University. It contains a collection of people images from Flickr images in "real" situations. By real, we mean that the pictures represent a variety of backgrounds, size and number of faces with different iluminations and rotations. They were ramdomly chosen from personal collections to build this dataset. The dataset is divided into several categories. We used only the "Group" category for our experiments. It contains $2,231$ photos of different groups of people with different backgrounds. The files have different sizes and resolutions. Some examples of images from the dataset is in Fig 4. We validated our method using this dataset.

B. Formulation

First experiment: The first experiment measures the behavior of variations of the image dimension and its detection times. Robustness was not analyzed in this experiment.

Four inteporlation kernels were chosen. The linear (LINEAR), the bi-cubic (CUBIC), the nearest-neighbor (NN) and

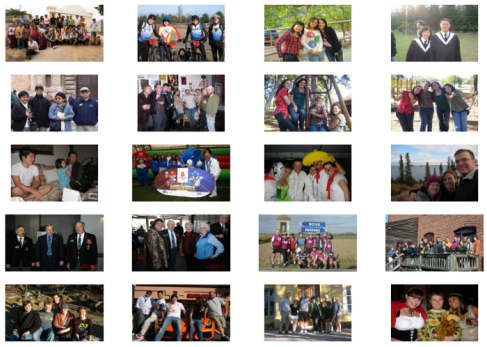

Fig. 4. 20 of 2231 images of the Cornell Dataset

the one based on pixel area relation (AREA). [9]

2576×1932 pixels JPG images were used with 5 faces to be detected, only for this experiment.

Second experiment: The second experiment explores the calibration method. A comparison between a non-scaling detection and performing our algorithm is made. Performance and robustness were analyzed.

The training image chosen was the *image_0328.jpg* from the Caltech dataset.

Third experiment: The third experiment explores the performance of our algorithm in a dataset with groups of different numbers of people. Using the Cornell dataset [8], we performed a series of detections over groups of these images. The images vary over illumintation conditions, rotations, filesize and resolution. We analyzed the impact of variances in our results.

VI. Results and Discussion

For all the experiments, we used the object detection framework proposed by Viola and Jones in [1]. The tests were implemented using the OpenCV Library [10] performing a face detection sequentially on every image from the datasets. The OpenCV Library also provides image resizing procedures with different interpolation kernels. All the tests were run on an Intel Core 2 Duo processor 2GHz, 3GB RAM and a Linux Kernel 2.6.32-32-generic 64-bits.

A. Performance

The relation between the scale and detection time has an exponential decay as we can see on the graph in Fig 5. The behavior of all the other interpolation kernels is quite similar.

From that, we can conclude that if the performance has to be improved on an object detection framework, there is always a limit for substantial gains. Values of scale s where $\phi(s)$ lies in the exponential "tail" will not improve the detection time substantially as long as s increases, even for different kernels.

The calibration method's output is presented in the Table I for the four interpolation kernels used. Average scales are calculated by choosing different $\epsilon \in [0.005, 0.150]$ as inputs for the calibration algorithm.

As expected by the results on Fig 5, the average detection times of the whole dataset for the 4 kernels are close to each other. This total average time is $2m15.979s$.

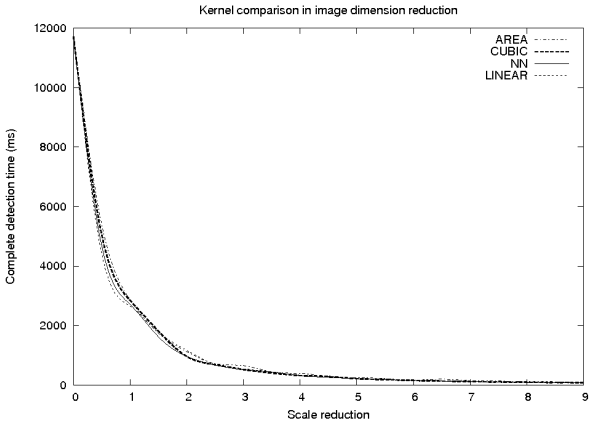

Fig. 5. Kernel comparisons

TABLE I
AVERAGE SCALES (LEFT) AND COMPLETE DETECTION TIMES (RIGHT)

Kernel	Average Scale
AREA	2.35
CUBIC	2.26
NN	2.42
LINEAR	2.28

Method	Detection Time
No-scaling	$8m28.998s$
Scaling	$2m15.979s$

TABLE II
NUMBER OF DETECTIONS FOR SCALES OF THE TABLE I

Kernel	0 faces	1 face	2 faces	3 faces	4 faces	5 faces
(none)	6	387	51	4	2	0
AREA	4	426	17	2	0	1
CUBIC	5	428	15	2	0	0
NN	4	430	15	1	0	0
LINEAR	4	429	17	0	0	0

The complete detection time using the calibration method is about 3.76 better than the detection without using scale reduction, as it is shown in Table 5.

B. Quality

We analyzed the robustness achieved by our algorithm. In the Caltech Database, all the images have only one face to be detected. The images can also provide false positives detections, that is verified by the first line of column II. The Viola and Jones framework without any dimension reduction had a total of $51 \times (2-1) + 4 \times (3-1) + 2 \times (4-1) = 65$ false positives detections.

As we can see in Table II, the calibration method was more robust than the classical no-scaling method. Among the 4 kernels analyzed, the LINEAR kernel achieved best results with a total of $17 \times (2-1) = 17$ false positives detections and only 4 images where none detection was found, representing a gain of 25.37% of robustness.

Table III shows how the robustness behaves on the variation of the scale. Note that when the scale is higher than 3, the number of detection misses increase, even though the number

TABLE III
SCALE IMPACT ON ROBUSTNESS FOR THE LINEAR KERNEL

Scale	0 faces	1 face	2 faces	3 faces	4 faces	5 faces
1.5	4	413	32	1	0	0
2.2	4	429	17	0	0	0
3.0	4	437	9	0	0	0
4.0	6	440	4	0	0	0
5.0	13	433	4	0	0	0
6.0	15	435	0	0	0	0

TABLE IV
DETECTION NUMBERS FOR SCALE 3 AND FOR THE CUBIC KERNEL

Scale	0 faces	1 face	2 faces	3 faces	4 faces	5 faces
3.0	6	437	6	1	0	0

of false detections decrease. In fact, the calibration method avoids "0 faces" detections over a small number of false detections.

The value of 3.0 for scale has a better result than the scale found by our calibration method: 2.2. On the other hand, the value 3.0 is closer to one of the limit of this closed interval $\phi^{-1}(1)$, 3.442 than 2.2. That means that the scale 3.0 is more likely to have loss of robustness than 2.2. In fact, for the CUBIC kernel, a scale of 3.0 results in a loss of 1 correct detection as shown in Table IV. Note that the "0 faces" detections increased by 1, showing a worse detection than the 2.2 scaling.

C. Groups of people

Our third set of experiments evaluated the performance of our method in a set of heterogeneous images. By heterogeneous, we mean images with different number of people, under different illumination conditions, rotations, filesize, etc.

As a first result, we noticed that the performance improves if we separate the dataset into groups with the same number of faces to be detected. If we perform our calibration method on these groups we calculate an optimized scale that better represents each group. For example, considering the whole dataset, the scale is equal to the minimum scale of the groups, which, in our results is 1. If we apply our method separately to each group with its own scale rate, we achieve better performance, and also better robustness.

In order to evaluate the impact of the variation of the scale, we used only image files larger than $140KB$. It represents more than half of the Cornell dataset and it showed improved gains. Small filesize images do not support a considerable resizing and therefore the gains are not substantial.

The Cornell dataset provides the eyes position of the faces of the images. So given face detection coordinates, we considered it a false positive (FP) if it does not surround a pair of corresponding left and right eye coordinates.

The dataset provides images of groups of $2, 3, \cdots, 37$ people. We used our method in images larger than $140KB$. The training image was the image with minimum distance between the eye coordinates provided. We claim that in real applications this step is human supervised. We used a calibration step $\epsilon = 0.01$.

TABLE V
RESULTS FOR THE GROUPS OF THE CORNELL DATASET WITHOUT NO PRE AND POST PROCESSING

#Faces	#Detections	Time	Detections	FP
2	70	$50.59s$	69	27
3	153	$80.43s$	108	35
4	1156	$468.43s$	1078	160
5	735	$239.75s$	659	69
6	564	$148.03s$	494	61
7	399	$87.23s$	350	31
8	472	$95.66s$	446	41
9	558	$98.37s$	532	71
10	340	$51.99s$	326	35
11	451	$59.27s$	420	30
12	468	$57.32s$	435	47
13	182	$20.41s$	158	8
14	350	$38.79s$	333	27
15	330	$33.39s$	317	18
16	192	$17.59s$	182	11
17	136	$11.98s$	122	4
18	198	$16.18s$	185	6
19	171	$13.39s$	157	10
20	180	$12.94s$	170	11
21	126	$8.90s$	115	11
22	66	$5.02s$	60	2
23	23	$1.51s$	19	1
24	48	$3.06s$	47	2
25	50	$2.94s$	47	1
26	52	$3.09s$	52	1
28	84	$4.71s$	83	9
29	87	$4.35s$	79	4
30	90	$4.12s$	71	0
33	66	$2.96s$	57	4
37	37	$1.29s$	24	2

Table V shows the results of not using any processing except for the Viola and Jones face detector. Table VI shows the results for our method. In both tables, each line represent a detection run over a group with same number of "#Faces". The total number of detection is given by "#Detections". These information is given in a metadata file by the Cornell dataset.

In most part of the groups, we achieved an increase of performance and robustness. For some groups there were no increase of performance. The training image, which contains the smallest faces of the group, could not be resized more than a $1 + \epsilon$ rate. For other groups (> 12), our method showed a small loss of robustness exposing a limitation. This is caused by the interference of false positive detections. During a resize process, false positives may appear. In our experiment, this situation was rare, occuring only 16 times over $7,260$ correct face detections. For groups of a large number of people, our method did not express substantial gains, most parts of the time, the appearance of false positives reduced the robustness. On the other hand, the performance for groups with small

TABLE VI
RESULTS FOR THE GROUPS OF THE CORNELL DATASET USING OUR METHOD

# Faces	# Detections	Scale	Time	Detections	FP
2	70	1.225	$37.95s$	70	23
3	153	1.125	$67.12s$	117	25
4	1156	1.105	$420.29s$	1080	133
5	735	1.03	$226.91s$	661	85
6	564	1	$148.90s$	494	61
7	399	1.05	$83.18s$	353	35
8	472	1.115	$77.29s$	472	38
9	558	1.135	$83.24s$	533	52
10	340	1.115	$41.20s$	340	29
11	451	1.01	$57.32s$	421	33
12	468	1.085	$50.17s$	430	39
13	182	1.025	$19.22s$	164	9
14	350	1.105	$31.65s$	333	31
15	330	1.075	$28.78s$	330	16
16	192	1.1	$14.48s$	181	12
17	136	1.05	$10.76s$	123	4
18	198	1.03	$15.15s$	187	8
19	171	1	$13.41s$	157	10
20	180	1	$12.99s$	170	11
21	126	1.145	$6.82s$	112	7
22	66	1.105	$4.15s$	59	1
23	23	1.06	$1.33s$	19	1
24	48	1.03	$2.81s$	47	3
25	50	1.055	$2.60s$	46	1
26	52	1.13	$2.41s$	52	52
28	84	1.075	$3.83s$	83	6
29	87	1	$4.41s$	79	4
30	90	1	$4.22s$	71	0
33	66	1.015	$2.91s$	56	3
37	37	1.015	$1.21s$	20	2

number of people per image were remarkable.

In sum, the total gain in performance was 10.26%. The increase of correct detections was only 0.34% and the reduction of false positives was 0.67%. Our method showed significant gains in performance maintaining a stable robustness validating our theory.

D. Limitations

The main limitation of our algorithm is the step of finding the training image $\bar{i}$ of a given image set I. This image is responsible for the maintenance of the robustness. Considering a heterogeneous set of images as we showed in our experiments, we only have significant gains if we split the set into groups of homogeneous images (same number of objects to be detected), otherwise the gains may not be considerable.

Another limitation is the definition of a general scale for groups of images with a big number of objects to be detected. The experiments showed that for groups of images with a large number of objects, there is no scale such that resizing every image can guarantee the robustness. This is due to image quality and resolution or even the influence of false positives.

Finally, a limitation in our approach is on the setup of the iterative step ϵ. Very small values can extend the running time of the algorithm in great proportions that could invalidate the use of this approach. We plan to find different solutions for finding the closed interval A_D and the mean value.

VII. CONCLUSION

A calibration method was presented for reducing the detection time of object detection framework. This method did not change in any aspect the framework itself. All the gains in performance were achieved by a pre and post processing. Our results showed, in most part of the cases, a gain of performance and a gain of robustness by supressing false positives. The comparison of 4 different 2D interpolation kernels was made for the proposed method using the Viola and Jones face detection framework. The LINEAR interpolation kernel was the most robust one with gains of 25% in detections performed in the Caltech Dataset. Our results using the Cornell Dataset, showed limitations in our method. Nevertheless, we achieve substantial gains in performance and robustness for most parts of the groups. In average, we achieve a gain of 10% in performance.

An application for future work is to study how a real-time object detection framework would adapt the calibration method presented in this work. Consider an enviroment where a user has a camera with a face detection software. The challenge for this kind of application is to define the training image $\bar{i}$ to calculate the best scale because there is no fixed image set I.

ACKNOWLEDGMENT

The authors would like to thank the Federal University of Minas Gerais, with special thanks to the Computer Science Department for supporting this work.

REFERENCES

[1] P. A. Viola and M. J. Jones, "Rapid object detection using a boosted cascade of simple features," in *CVPR (1)*, 2001, pp. 511–518.

[2] P. Viola and M. Jones, "Robust real-time face detection," in *International Journal of Computer Vision*, vol. 57, no. 2, 2004, pp. 137–154.

[3] M. A. Anjum, M. Y. Javed, and A. Basit, "Face recognition using double dimension reduction," in *WEC (2)*, 2005, pp. 43–46.

[4] P. Burt and E. Adelson, "The laplacian pyramid as a compact image code," *Communications, IEEE Transactions on*, no. 4, pp. 532–540.

[5] H. Gao, H. K. Ekenel, and R. Stiefelhagen, "Robust open-set face recognition for small-scale convenience applications," in *DAGM-Symposium*, 2010, pp. 393–402.

[6] Z. Wang and Z. Miao, "Scale-robust feature extraction for face recognition," in *17th European Signal Processing Conference (EUSIPCO 2009)*, 2009.

[7] M. Weber. (1999) Caltech frontal face database. [Online]. Available: http://www.vision.caltech.edu/html-files/archive.html

[8] A. Gallagher and T. Chen, "Understanding images of groups of people," in *Proc. CVPR*, 2009.

[9] T. M. Lehmann, C. Gönner, and K. Spitzer, "Survey: interpolation methods in medical image processing." *IEEE transactions on medical imaging*, no. 11, pp. 1049–1075, Nov.

[10] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.