Representation of Pose Invariant Face Images Using SIFT Descriptors

1st Nthabiseng Mokoena
Modelling and Digital Science
Council for Scientific and Industrial Research (CSIR)
Pretoria, South Africa
NMokoena1@csir.co.za

2nd Dr. Kishor Nair
Modelling and Digital Science
Council for Scientific and Industrial Research (CSIR)
Pretoria, South Africa
KNair@csir.co.za

***Abstract*—The choice of a face database should solemnly depend on the problem to be solved. In this research work, we use the Face Recognition Technology (FERET) database to address the challenge of face pose variations. The Scale Invariant Feature Transform (SIFT) is used to represent these face images in the database. SIFT has been proven to be a robust and a powerful method for general object detection in the past years. This method is now popular in the field of face recognition for purposes of extracting key points which are scale and orientation invariant from the face image. This work demonstrates that through extracting SIFT features from different face image patches and at different sigma σ values, a face pose can be classified towards better pose invariant face recognition.**

***Index Terms*—Face recognition, Pose-invariant face classification, SIFT, Machine Learning algorithms**

I. Introduction

Naturally, the main objective of a Face Recognition Technology (FRT) is to identify or verify a scanned frontal profile face of a living person, by matching it with a set of pre-stored database faces [1], [2]. These images are taken under controlled environment, in terms of illumination, pose and expressions. FRT is a form of biometric technology which is becoming popular in access control, law-enforcement and at the airport. While face recognition in frontal images remains a well-studied area [3]–[5], recognition of faces acquired with changes in illumination, pose and expression is still a challenge in FRT. During the previous years, researchers in the field of computer vision and machine learning, have been developing algorithms that try to solve the problem of recognition in a non-frontal view [3]. These kind of algorithms are called pose-invariant face recognition (PIFR) algorithms [4]. The main purpose of PIFR algorithms is to solve the difficulty of a FRT to identify a person who is in a non-frontal view. In this research work and other research works reviewed in this paper, the word 'pose' refers to the position of a face in the image, in terms of degree angles (°).

There are a number of challenges that a pose invariant image create, which may affect the performance of a FRT, such as, self-occlusion [4]. In self-occlusion there is loss of information on the face image, which may lead to loss of semantic correspondence of images, further leading to difficulties in recognising faces. Techniques to solve the problem of face pose includes extracting pose invariant features, which may be divided into two categories, namely, engineered features and learning based features [4]. Furthermore, engineered features maybe subdivided into, landmark detection based methods and landmark detection-free techniques.

Landmark detection methods have been used in order to learn semantic correspondence of images that have pose variations [3], [4]. These type of features can be extracted manually or using machine learning models. Using the SIFT algorithm to extract features on the selected landmarks, the main purpose of this research work is to represent a face in a compact feature vector of 128 keypoints, also to classify pose invariant face features by machine learning models in order to determine the pose class of a face. During the late 90's, Lowe [6] developed an object recognition algorithm called the Scale Invariant Feature Transform (SIFT). The SIFT algorithm transforms images into invariant local feature vectors. These feature vectors are invariant to scaling, rotation and translation [6]. There are four (4) main steps that Lowe describe in creating these scale invariant features [14]. The first step is scale extrema detection using Difference-of-Gaussian (DoG) function. The algorithm performs scale spaces by creating progressive blur on an image, then re-size the image to half of the original size. The process is repeated according to the number of octaves (image of the same size but different scales) selected. The second step is keypoints localisation. The process of locating keypoints is divided into two parts, a. Locating maxima and minima in the DoG, and b. locating subpixel maxima and minima. The third step is keypoints orientation assignment, this step assigns orientation to the tested scale-invariant keypoints from the previous steps, in order to collect the magnitude and gradient of these keypoints. The last step is to generate the keypoint descriptor. The aim of this step

is to generate a unique descriptor which is easy to calculate and compare to other keypoints. In recent years, extracting scale invariant features is becoming popular in the field of face recognition. The next section will discuss some of the research work in pose invariant face representation.

II. Related Work

Methods of extracting engineered features, in order to correct pose using landmarks started in the early 1990's with [8] and [9]. In their research work, Brunelli et al [9] proves that a face can still be recognised even when face features such as the nose, mouth and eyes are not visible. They represent a face by extracting relative position (geometrical information) and other parameters of the selected landmarks such as the tip of the nose, eyes and chin. Extending this geometric methods for representing a pose invariant face, [11], [12] use epipolar lines and canonical-view image selection respectively. In these methods, a representative image for each identity to be trained is selected in order to learn the transformation between these face images, using Labeled Faces in the Wild dataset (LFW). To learn the transformation of these images they build a componet-based convolutional nueral network. The pose robust face is represented canonical-view images.

The SIFT algorithm is used in [13], [14] to represent pose invariant faces at landmark level. Biswas et al [13] method transforms the extracted features from poor quality input image to approximate the high quality gallery/database images using Multidimensional Scaling. To represent a face for features transformation and recognition, they extract local SIFT descriptors at the corner of the eyes, corner of the mouth and the tip of the nose. These low resolution features are transformed to a space in which their inter-Euclidean distances is similar to the high resolution images in the gallery [13]. In their proposed method they used the Multi-PIE database and their method showed that representing a low resolution image with SIFT descritors, face images can still be recognised.

Other methods that have been used in recent years to represent a pose invariant face image include Local Binary Patterns (LBP) [15], [16] and the dual-cross patterns [17]. Chen et al [15] in their method, they represent a pose invariant face by extracting Local Binary Patterns and demonstrate state-of-the-art performance of the high dimensional LBP descriptors. Firstly, they locate the five facial landmarks that they selected using explicit shape regression [18], i.e. mouth corners, nose tip and eyes corners. Then they constructed a multi-scale LBP descriptor based on these landmarks. Each patch is then divided into a grid of cells, and code it by a descriptor [15]. Then combine all LBP descriptors to form one high dimensional feature which represent a face.

III. Proposed Method

This section explains the formal steps that lead to the method proposed for this research work. The methodology of this study is based on the model of Ho et al [10] for classification of face pose. Contrary to [10], the purpose of this study is not to use Markov Random Fields (MRF's) for matching face images, but the purpose is to implement a classification model that will be able to classify a still face image according to pose. Additionally, Fig. 1 illustrates a flow chart of the proposed methodology, the succeeding sections will explain the steps in details.

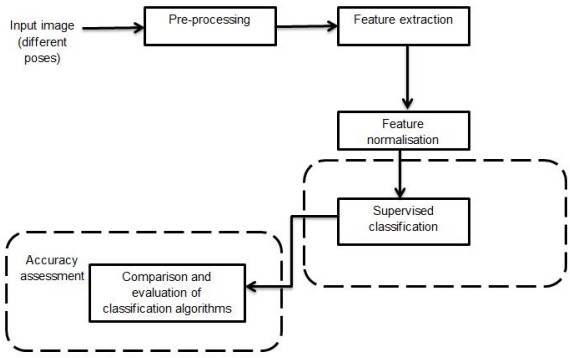

Fig. 1. Proposed methodology.

A. Dataset

The dataset used in this research work are randomly selected mugshots from the Facacial recognition Technology (FERET) database. The FERET database is distributed by the National Institute of Standards and Technology (NIST). The purpose of the distribution of this database is to help researchers to develop automated face recognition algorithms, technology and techniques. These mugshots were selected by going through the database and selecting images that had all the required poses per subject. The excluded mugshots were those which had some or none of the poses we selected for the training set. Selected mugshots were from subjects from different races, i.e. Blacks, Whites and Asians. Some of these subjects have wrinkles, facial har (bearded) and some have their eyes half open. Fundamentally, these subjects are in four selected poses, i.e. $\pm$22,5°and $\pm$ 45°, i.e. P1 = -22,5°, P2 = +22,5°, P3 = +45°and P4 = -45°. Fig. 2 shows some of the images that were used as dataset from the FERET database.

B. Preprocessing

The main purpose of pre-processing process is to enhance or improve the image, in usual cases, the enhanced image would be presented in pixels. The enhancement process is done to remove data on the image that causes distortion, for both input and output images. Pre-processing process in this research work involves 3 (three) steps, namely, patch cropping, pixel brightness transformation and image filtering.

1) Patch cropping: The montage of images in Fig. 1 visibly shows that the images are relatively large (mugshots showing sholder area), to be directly processed. As such, we select only the local landmarks, i.e. eyes, nose and mouth, as opposed to using the whole face area (global). Additionally, because the

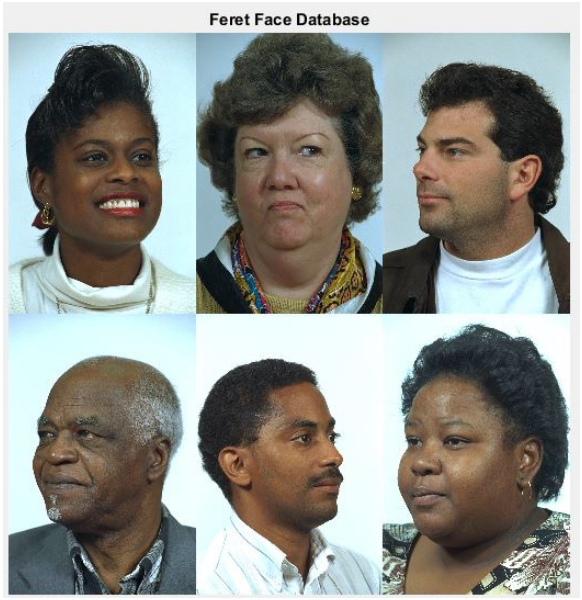

Fig. 2. Different face poses from the FERET database.

images are not of the same size, the patch size is not fixed, the size varies from patch to patch, i.e. one subject's mouth patch size is not the same as the next subject. The main idea is to obtain a rectengular shape patch that covers the area of the chosen landmark. Algorithm 1 shows a psudo code on how patches were selected. Fig. 3 shows the different eyes, nose patches of different subjects that built the dataset.

Algorithm 1 Patch cropping

1: *face image* ← *PatchSize*
2: **if** *PatchSize* = m x n matrix (width × height) **then** $imageCropping$ = *save patch*
3: **else** *imageCropping* = *discard patch*
4: **end if**

Fig. 3. Eyes, nose and mouth patches.

2) Pixel britness transformation: The motive behind this process is to give patches a uniform pixel brightness. Using grey-scale image transformation, the main aim is to attain images with equally distributed brightness levels over the whole brightness scale. Furthermore, greyscale transformation retains the illuminance of the image, without changing the pixel position. The distinction between black and white (BW) and grey-scale images is that BW have only two pixel colours, i.e. black and white pixels; whereas, grey-scale intensity is stored as an 8-bit integer, i.e. 256 possible shades of grey from black and white. The transformation function is given linearly as (1):

$$s = T(r) \tag{1}$$

where s is the pixel of the output image, r is the pixel of the input image and T is a transformation function that maps each value of r to each value of s.

3) Image filtering: One of the reasons we choose this method of filtering for the patches, is because it allows edges to be preserved. Considering that for the next process (feature extraction), image edges need to be computed. Additionally, this method of image filtering has not only proven to have better behaviours near edges, it can also transfer the structure of the guidance image (be it the input, output or a different image). The filtering output patch at a pixel i is a weighted average of (2):

$$q_i = \sum_j W_{xy}(I)\rho_j \tag{2}$$

where x and y are the pixels at a given point, and I is the guidance image whose function is a filter kernel W_{ij} and is independent of ρ.

C. Feature Normalisation

The normalisation procedure used in this research work is the Linear scaling to unit variance method. The main purpose of choosing this procedure over others is to transform the feature component x to a random variable with mean 0 and unit variance. The procedure taken to achieve scaled features is described in subsequent sections:

1) Calculate the Mean (μ): Calculate the Mean (μ) means we calculate the average of all the feature vectors extracted from each patch. We have 100 patches per pose, i.e. P1 - P4, the mean is calculated per pose as the average of features (100 x 128), 100 images each having 128 feature vectors. We then use (3):

$$\bar{X}_j = \frac{\sum_{i=1}^{n} X_i j}{n} \tag{3}$$

where:

- n is the keypoint of all the patches, n = 100;
- i is each keypoint in a feature vector per patch (column), such that i = 1, ..., n;
- j is each keypoint in a feature vector per patch (row), j = 1, ..., 128;
- X_{ij} is the extracted keypoint at position $_{ij}$.

In other words, the mean μ of a pose class will be a feature vector of 128 elements representing the mean of any given class of the four classes we have. The purpose of this step is to find the average per pose class. The next step is to calculate the spread of features per class.

2) Calculate the Standard Deviation (σ): The definition of Standard Deviation is given as, the average distance from the mean of the data set to a point. Here we calculate how spread out our features are, by calculating the squares of the distance from each data point, to the mean of the set. Thereafter, we added all the calculated squares, divided by $n-1$, then took the positive square root. Equation (4) is:

$$s_j = \sqrt{\frac{\sum(X_{ij} - \bar{X_{ij}})^2}{n-1}} \tag{4}$$

Where:

- n is the keypoint of all the patches, n = 100
- i is each keypoint in a feature vector per patch (column), such that i = 1, ..., n
- j is each keypoint in a feature vector per patch (row), j = 1, ..., 128
- s_j is the keypoint at the j_th position
- X_{ij} is the raw feature
- $\bar{X}_{ij}$ is the mean at a position

After determining the mean and standard deviation of the four poses, we then calculate the normalised feature vector for each patch using (5):

$$\tilde{x_i} = \frac{|x_i - \mu_j|}{\sigma_j} \tag{5}$$

In the above equation, we let:

- x_i be the extracted keypoint at the i^{th}position;
- μ_j be the calculated mean at the j^{th} position;
- σ_j be the calculated standard deviation at the j^{th} position;
- $\tilde{x_i}$ be the normalised keypoint at the i^{th} position.

D. Feature extraction

Our feature extraction method uses the original SIFT feature extraction. However, while the original scale parameter (σ value) of SIFT is zero (0), our method extracts the SIFT keypoints at three different scale parameters; σ = 3, σ = 6 and σ = 9. This is done so as to observe the value of σ that allows extraction of enough reliable keypoints, for better classification of face pose, i.e. the level of blur can give us better classification of face images at different poses. Furthermore, in order to create a 128-dimension descriptor, we keep the original window at 16 x 16 around the keypoint and divide the window into sixteen 4 x 4 windows. Therefore, the magnitude ($m(x_i, y_i)$) and orientation ($\theta(x_i, y_i)$) of each patch sample (pixel) will be calculated for all classes ($\pm 22,5°, \pm 45°$). The objective of changing the scale factor (σ) is to extract pose invariant features, by locating meaningful keypoints and constructing significant descriptors which can also be used for texture classification in future research work.

E. Supervised Classification

To train the dataset, we use a multiclassification methods. For each training observation, there exist a corresponding class label, $y_i \in \{1, -1\}$, represented by Y which is an array of class labels (P1 - P4) where each row corresponds to the value of corresponding x_i. Therefore, n normalised SIFT feature vectors of dimension 128 that contains training observations of $(\vec{x_i}, y_i)$ which represent the feature at position i and the corresponding class label at y_i. Furthermore, test observations which will be used at a later stage in order to test the classification performance of our model are represented by $\vec{x}^* = (x_1^*, ..., x_p^*)$. With this information, our goal is to classify into which pose (°) the input image falls.

IV. Experiments

Upon obtaining the patches using Algorithm 1 from the previous section, pre-processing is performed by firstly, re-sizing all patches to the same size, i.e. (200 x 200 (wxh)), in order to be able to combine them. The re-sizing will further be useful when we extract features, as any size lower than 200 x 200 was not sufficient for extraction of features for some patches. The next step is to perform image filtering method applying equation 2, we obtain an image as shown in Fig. 4. This image is a patch for one subject in which features will be extracted in order to form a representation, this is the reason why we combined the patches. From the filtered image, the SIFT keypoints are extracted at different scale parameters (σ), as a result, creating different scale spaces, i.e. scale space are created at (σ = 3, σ = 6 and σ = 9) for each pose. The choice of different scale parameters is performed so as to identify the ideal scale space for our dataset.

A. Classification at σ = 3

Features were extracted and trained at σ = 3, normalised using linear scaling to unit variance. These features were trained using distance-based measures. The Hamming distance metric shows a better classification rate as compared to other distance metrics that were trained (Minikowski (cubic) and Cosine), obtaining a 98,8% classification rate, this means that the likelihood that a face pose will correctly be classified for the trained pose is high. Nonetheless, classification rate on its own does not explain much about the classified data. To get a clearer overview about the performance of the classifier, we then used confusion matrix of a size 4 x 4 matrix. The confusion matrix shows that for both Pose 1 and 2 the classifier predicted 99 features correctly. For True Positives (TP), the classifier prediction rate was 99% , only 1% was misclassified False Negative (FN) as pose three for both classes. Similarly, the confusion matrix also shows that 98 features from Pose 3 and 4 were correctly classified, corresponding to the 98% and 4 of the features were misclassified as pose 3 and 4 respectively.

B. Classification at σ = 6

Features which were extracted at σ = 6 were better classified using decision trees. With maximum of 10 splits and 10 surrogates, we obtained a classification rate of 97%. The observation that was made is that, even though we increased the number of splits, we still obtained the same classification percentage. Fig. 4 shows a plot of the ROC curve for the trained tree, where we plot the TP (y-axis) against the FP

(x-axis). The interpretation is that the closer the curve falls to the y-axis and to the top line of the ROC space, the better classification accuracy the trained classifier gives. The Confusion matrix for the decision tree shows that for Pose 1, 95 features were correctly classified, which corresponds to the 95%. However, three of the 100 features were misclassified under Pose 3 and 2 were misclassified under Pose 4. For Pose 2, 99 features were correctly classified, and one feature was misclassified under Pose 1. Three features which belong in Pose 3 and two features which belong in Pose 4 were also misclassified under Pose 1.

C. Classification at σ = 9

An SVM classifiergave better classification rate for features which were extracted at σ = 9. The kernel scale of the classifier was adjusted depending on the classifier's performance. At kernel scale 4 we obtained a classification rate of 72%, any number above caused a decrease in the classification rate. Even though we used 25 folds of validation division, the plot from the obtained ROC curve, showed that the model was not learning very well, as the graph was not as smooth as the previously discussed data, where the sigma value was low. At the same time, the confusion matrix in Fig. 5, also shows (in red blocks) a lot of misclassified classes, which proves that for features extracted at σ = 9, our data set is not performing very well, even after training the data set with the previous classifiers (decision tree and k-NN) the performance was still unsatisfactory.

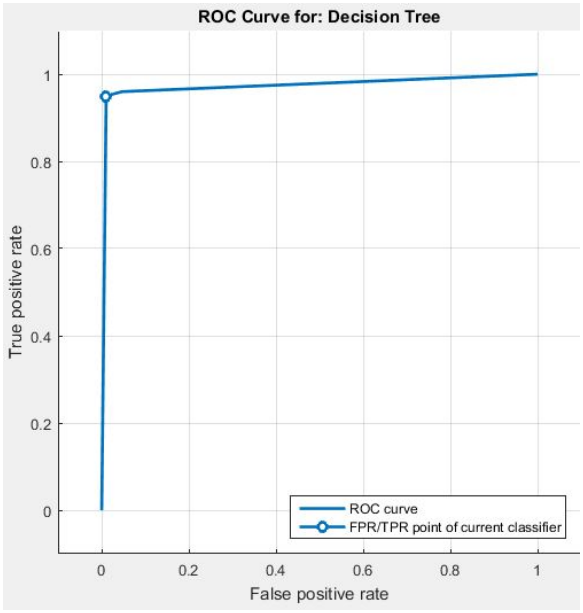

Fig. 4. ROC curve at σ = 6.

V. Evaluation of Classification Algorithms

In most cases, the confusion matrix used above is useful to understand the performance of a classifier given a dataset. However, the confusion matrix is not a model evaluation

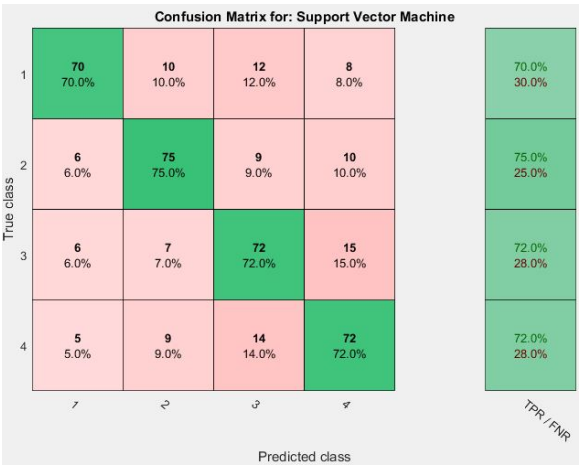

Fig. 5. SVM confusion matrix at σ = 9.

matric. To be able to choose a better performing classifier. For the model evaluation, we used two performance metrics, Sensitivity (True Positive Rate) and Specifisity (True Negative Rate) using (6):

$$Se = \frac{|TP|}{|TP| + |FN|}$$
$$Spe = \frac{|TN|}{|FP| + |TN|} \quad (6)$$

Initially, evaluation of Specificity and Sensitivity was performed for each class, i.e. one against all. Using the information from each class, the second step was to calculate the overall performance of a given classification model, using real numbers not percentages. At σ = 3, Sensitivity was 0.98 and Spe = 0.98. Whereas, σ = 6 obtained Se = 0.94 and Spe = 0.97. Lastly, σ = 9 obtained Se = 0.64 and Spe = 0.64.

VI. Conclusion

Face recognition extracting SIFT features is a relatively well-researched area, although many difficulties and challenges still remain. However, in this paper, the objective is to classify the pose of an image. The well-known FRETE database has been used for this paper. We believe that identifying the pose first can help in more accurate face recognition. From the classification rates and evaluations obtained in the previous sections, the conclusion is that kNN performs better where σ = 3. While on the other hand, SVM at σ = 9, gave us undesirable classification rate, with the features that were extracted. The classification model may be useful in real world face recognition applications, as a pose classifier before a face can be recognised.

References

[1] W. Zhao, R. Chellappa, P.J. Phillips and A. Rosenfeld, "Face recognition: A literature survey," ACM computing surveys (CSUR), vol.35, pp.399-458, 2003.

[2] A.F. Abate, M. Nappi, D. Riccio and G. Sabatino, “2D and 3D face recognition: A survey,” Pattern recognition letters, Elsevier, vol.28, pp.1885-1906, 2007.
[3] M.A. Turk and A.P. Pentland, “Face Recognition Using Eigenfaces,” CVPR’91 IEEE Computer Society Conference, 1991.
[4] C. Ding, and D. Tao, “A comprehensive survey on pose-invariant face recognition,” ACM Transactions on intelligent systems and technology (TIST), vol. 7, p.37, 2016.
[5] R. Chellappa, C.L. Wilson and S. Sirohey, “Human and machine recognition of faces: A survey,” Proceedings of the IEEE, vol. 83, pp.705-741, 1995.
[6] D.G. Lowe, “Object Recognition from Local Scale-Invariant Features,” in Computer vision. The proceedings of the seventh IEEE international conference, 1999.
[7] D.G. Lowe, “Distinctive image features from scale-invariant keypoints,” International journal of computer vision, vol. 60, pp.91-110, 2004.
[8] B. Roberto and T. Poggio, “Face recognition: Features versus templates,” IEEE Transactions on pattern Analysis and machine Intelligence, vol. 15, pp.1042-1052, 1993.
[9] A. Pentland, B. Moghaddam and T. Starner, “View-based and modular eigenspaces for face recognition,” Vision and Modeling Group, Media Laboratory, Massachusetts Institute of Technology, 1994.
[10] H.T. Ho and R. Chellappa, “Pose-invariant face recognition using markov random fields,” IEEE transactions on image processing, vol. 22, pp.1573-1584, 2013.
[11] C.D. Castillo and D.W. Jacobs, “Using Stereo Matching for 2-D Face Recognition Across Pose,” CVPR’07. IEEE Conference, 2007.
[12] Z. Zhu, P. Luo, X. Wang and X. Tang, “Recover canonical-view faces in the wild with deep neural networks.” arXiv preprint arXiv:1404.3543, 2014.
[13] S. Biswas, G. Aggarwal, P.J. Flynn and K.W. Bowyer, “Pose-robust recognition of low-resolution face images.” IEEE, vol. 35, pp.3037-3049, 2013.
[14] D.G. Lowe. “Distinctive image features from scale-invariant keypoints.” Springer, vol. 60, pp.91-110, 2004.
[15] D. Chen, X. Cao, F. Wen and J. Sun, “Blessing of Dimensionality: High-Dimensional Feature and its Efficient Compression for Face Verification,” IEEE, 2013.
[16] M. Guillaumin, J. Verbeek and C. Schmid, C. “Is That You? Metric Learning Approaches for face identification,” IEEE 12th international conference, 2009.
[17] C. Ding, J. Choi, D. Tao and L.S. Davis, “Multi-directional multi-level dual-cross patterns for robust face recognition.” IEEE, vol. 38, pp.518-531, 2016.
[18] X. Cao, Y. Wei, F. Wen and J. Sun, “Face alignment by explicit shape regression.” International Journal of Computer Vision, Springer, vol. 107, pp.177-190, 2014.