The following publication Hailiang Li, Man-Yau Chiu, Kangheng Wu, Zhibin Lei, and Kin-Man Lam "Accurate car plate detection via car face landmark localization", Proc. SPIE 10420, Ninth International Conference on Digital Image Processing (ICDIP 2017), 1042012 (21 July 2017) is available at https://doi.org/10.1117/12.2281924.

Accurate Car-Plate Detection via Car Face Landmark Localization

Hailiang Li*[a], Man-Yau Chiu[a], Kangheng Wu[b], Zhibin Lei[b], Kin-Man Lam[a]
[a]Department of Electronic and Information Engineering, Hong Kong Polytechnic University
[b]Hong Kong Applied Science and Technology Research Institute Company Limited, China

ABSTRACT

For intelligent vehicle surveillance systems, it is a big challenge to detect small, blurred car plates of vehicles driving on a highway. In this paper, we present a novel, two-stage detection scheme for small, blurred car-plate detection in large surveillance images. Our proposed scheme firstly detects vehicles, and then locates the car plates in specific regions of detected vehicles based on our proposed car-face landmark localization algorithm. Our scheme can also solve the high false-alarming rate problem with small, blurred car-plate detection. Experimental results show that our proposed method is accurate, and able to reduce the false-alarming rate, without any compromise in speed.

Keywords: Vehicle detection and tracking; Car-plate detection; Intelligent traffic; Surveillance System;

1. INTRODUCTION

Car-plate detection is one of the most critical steps on intelligent traffic applications, and is important for car-plate recognition. For automatic licence-plate detection, researchers have used many diverse methods, including corner template matching [1], Hough transform [2], histogram analysis [3], morphological operations [4], and sophisticated mechanisms, such as vector quantization [5], neural networks [6], etc. However, most of the car-detection algorithms used are based on genetic object-detection methods, or based on methods for other domains, such as face detection. Viola et al. [7] proposed an efficient method for face detection, in which the integral-image representation was employed for computing the Haar-like features [8], and the AdaBoost learning algorithm was used for learning a fast and accurate detector. The AdaBoost learning algorithm, first proposed by Freund and Schapire [9], selects simple Haar-like features from a complete feature set to form weak classifiers, and builds a strong classifier by combining the weak classifiers. Based on the AdaBoost algorithm, a licence-plate detection method was proposed in [10], which is rapid and robust against variation in illumination. Instead of using one stage to deal with all situations, the proposed method in [10] consists of two stages: one for dealing with normal illumination, the other for dealing with abnormal illumination.

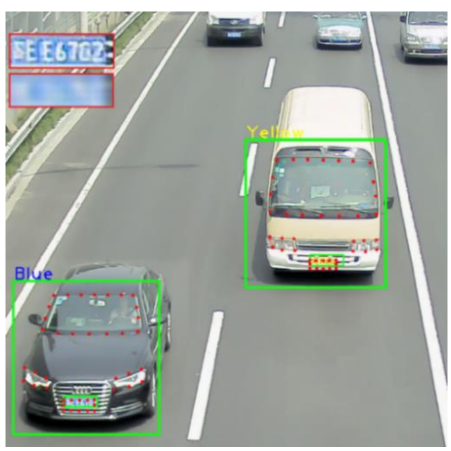


Figure. 1: A scene of small, blurred car-plates detection and car-plate color extraction in a high-way surveillance system

*harley.li@connect.polyu.hk,{harleyli, edmondchiu, khwu, lei}@astri.org, enkmlam@polyu.edu.hk; www.astri.org

Although there have been lots of successful traffic applications based on car-plate detection, most of them carry out the detection at gates on highways or car-park entries, where the numbers and characters of a car plate can be extracted much more easily. It will become much more challenging for detecting small, blurred car plates in a highway or street situation, as with the scene shown in Figure 1. Furthermore, the resolution of the captured images from surveillance cameras is usually very large (e.g., 1920×1080 pixels), but the size of car plates in the images is very small and blurred. This makes the detection involve high computational complexity, and result in many false positives.

The motivation of our study on detecting small, blurred car plates in highway surveillance systems is that the color of a car plate can provide useful information about the vehicle detected. For example, in mainland China, licence plates are issued by China's Vehicle Management Offices, under the administration of the Ministry of Public Security. The color of a car plate implies specific meaning. Yellow plates are issued for larger vehicles, such as trucks, buses, etc., while blue plates, the most common type, are issued for small or compact vehicles. Therefore, in some situations, where the car plates are too blurred for car number recognition, extracting the car-plate color can still provide useful information for intelligent traffic control. The color information can also assist in vehicle counting and car-type identification. To this end, we have formulated a unified and generalized framework with three innovations. (1) We devised a two-stage car for high-speed car-plate detection. (2) We proposed a set of car-face landmark points for car-face localization. (3) We developed a robust blurred car-plate detection algorithm, based on the AdaBoost detector, constrained by the car-face landmarks located.

The remainder of this paper is organized as follows. Firstly, we will review the random-forest-based car-face landmark localization algorithm and our proposed car-face landmark points in Section 2. In Section 3, we will describe our two-stage scheme for car-plate detection, and our proposed small, blurred car-plate color extraction method. Section 4 will present the experiment results, and a conclusion is given in Section 5.

2. RANDOM FORESTS FOR LANDMARK LOCALIZATION

2.1 Motivation on locating landmarks for car-faces

Landmark localization algorithms, which extracts features from the neighborhood of facial landmarks in face images, were first employed for face alignment and recognition. Usually, 17, 29 or 68 facial landmarks, which are located around the eyes, nose, lips, or the jaw, as shown on the left in Figure 2, are selected. These landmark locations carry the most amount of semantic information about faces, which is useful for discriminative and generative approaches for facial-image analysis. For face alignment, the required landmarks are located by using supervised machine learning methods, whereby a model is trained from a large amount of human-labeled face images. The learned model can then be used for facial shape estimation on unseen images, which contain face(s).

To our knowledge, there is no landmark localization applied to images with vehicles, until now. Motived by face-landmark localization, we propose using 42 landmarks for car-faces, as shown on the right of Figure 2. These landmarks are used to represent the windscreen, headlights, and number plate, which are the most commonly seen components in the front face of a car, and have consistent relative positions to each other. Our method was also inspired by the animated film "Cars" presented by Walt Disney Pictures, where the cars were animated as live cars with mouths and eyes in their frontal car-faces.

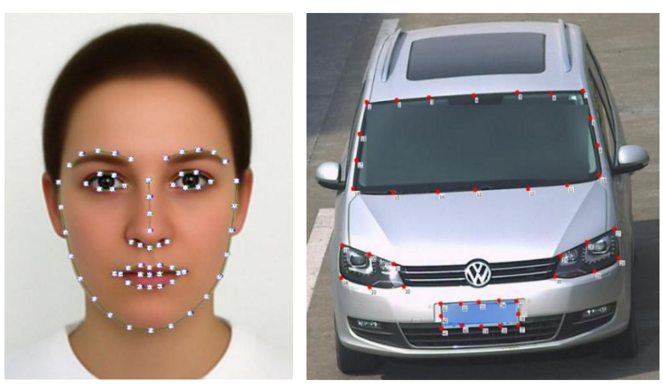


Figure. 2: Landmarks on a human face and a car-face

2.2 Random Forests

There are mainly two approaches for face alignment - Active Appearance Models [14], which build parametric models for facial appearance, and regression-based models, which directly model the mapping from appearance to facial shape. There are a number of state-of-the-art methods for face alignment, but we consider the regression-based methods, based on using a cascaded shape-regression framework, first proposed in [12]. Different from other methods, the regression-based approach progressively refines an initial shape estimate, in several stages, directly from appearance, without learning any parametric shape or appearance models. Random forests can be used for the estimation. A random forest is an ensemble of T binary decision trees, $T^t(x): V \to R^K$, where t is the index of the trees, $V \in R^M$ represents features in the M-dimensional feature space, and $R^K = [0, 1]^K$ is the space of class probability distributions over the label space $Y = \{1, . . . , K\}$. In the testing/prediction stage, each decision tree returns a class probability $p_t(y|\boldsymbol{v})$ for a given test sample $\boldsymbol{v} \in R^M$, and the final class label y^* is then obtained, via averaging, as follows:

$$y^* = \arg\max_{y} \frac{1}{T} \sum_{t=1}^{T} p_t(y|\boldsymbol{v}). \tag{1}$$

A splitting function $s(v;\Theta)$ is typically parameterized by two values: (i) a feature dimension $\Theta_1 \in \{1, \ldots, M\}$, and (ii) a threshold $\Theta_2 \in R$. The splitting function is defined as follows:

$$s(v;\Theta) = \begin{cases} 0, & \text{if } v(\Theta_1) < \Theta_2, \\ 1, & \text{otherwise,} \end{cases} \tag{2}$$

where the outcome determines the child node where the sample $\boldsymbol{v}$ will reside, and the labels 0 and 1 imply that the sample belongs to the left and the right child node, respectively. Each node chooses the best splitting function Θ^* from a randomly sampled set $\{\Theta^i\}$, by optimizing the following function:

$$I = \frac{|L|}{|L|+|R|} H(L) + \frac{|L|}{|L|+|R|} H(R), \tag{3}$$

where L and R are the sets of samples that are routed to the left and the right child node, respectively, and $|S|$ represents the number of samples in the set S. During the training of a random forest, each decision tree is provided with a random subset of the training data (i.e. bagging), and the trees are trained independently of each other. Training a single decision tree involves recursively splitting each of its nodes, such that the training data in the newly created child nodes are clustered according to class labels. Each tree is grown until a stopping criterion is reached (e.g. number of samples in a node is less than a threshold or the tree depth reaches a maximum value), and the class probability distributions are estimated in the leaf nodes. $H(S)$ is the local score for a set of samples (S is either L or R), which is normally calculated using entropy measure as in (6), but it can be replaced by variance [12, 15] or the Gini index.

$$H(S) = -\sum_{k=1}^{K} [p(k|S) * \log(p(k|S))] \tag{4}$$

where K is the number of classes, and $p(k|S)$ is the probability for class k, which is estimated from the set S.

2.3 Cascaded shape regression

Many face alignment methods work under a cascaded framework, where an ensemble of N regressors operates in a stage-by-stage manner, and are referred to as stage regressors. This approach was first explored in [8]. At the testing stage, the input to a regressor R_t at stage t is a tuple (I, S^{t-1}), where I is an image and S^{t-1} is the shape estimate from the previous stage (the initial shape S^0 is usually the mean shape of the training samples). The regressor extracts features with respect to the current shape estimate, and regresses a vector of shape increment as follows:

$$S_t = S_{t-1} + R_t\big(\phi_t(I, S_{t-1})\big), \tag{5}$$

where $\phi_t(I, S_{t-1})$ is referred to as the shape-indexed features, i.e. they depend on the current shape estimate. The cascade progressively infers the shape in a coarse-to-fine manner – the early regressors handle large variations in shape, while the later ones ensure small refinements. After each stage, the shape estimate resembles the true shape more closely. In our algorithm, the feature mapping function $\phi_t(I, S_{t-1})$ generates the local intimacy definition feature (IDF) [15] values derived from the shape-indexed feature. There is an assumption, proved by intensive experimental results, that the shape increments have close correlation with the local features of the landmarks, which define the face shape. Thus, given the features and the target shape increments $\{\Delta S_t = S - S_{t-1}\}$, we can learn a linear projection matrix R_t. Most cascaded regression models [12, 15] share a similar workflow, as shown in Figure 3.

3. A TWO-STAGE SCHEME FOR CAR-PLATE DETECTION

In this paper, we propose a robust scheme for extracting small, blurred car-plates from surveillance images. There are two challenges for this task: detection at high speed and accuracy requirements with small and blurred images. Our algorithm will first detect a small car-plate in a large image frame (1920×1080 pixels), while keeping the video capturing in real time (i.e. 30 frames per second), which is a difficult, computational task. Conventional methods usually employ foreground and background segmentation to identify the possible car areas, which is simple but not a robust scheme. Since cars, which can be detected robustly, are relatively larger and clearer objects in images, a more robust scheme based on two stages is proposed. The first stage will detect car candidates using the AdaBoost algorithm, then the second stage detects car-plates in the respective car-region candidates. Experimental result show that our two-stage car-plate detection algorithm can work more efficiently and robustly.

Although experimental results have proven that AdaBoost with Haar-like features can detect clear car-plates robustly. However, for the task of detecting small and blurred car-plates, the method has achieved unsatisfactory performance, in terms of both the detection rate and false alarming rate. The HOG feature has achieved good and robust performance on pedestrian detection [11], which is based on the fact that object appearance and shape can often be characterized well by the distribution of local intensity gradients or edge directions. Therefore, this feature can still represent the shapes of very blurred car-plates. Similar to pedestrian shapes, and motivated by [11], we propose using the HOG feature to replace the Haar-like feature for the detection. Our experimental results show that the AdaBoost+HOG method can effectively detect car-plates with a very low missing rate, and with a very high false positive rate. Currently, if the false alarming problem could be solved, then an efficient and accurate car-plate detection algorithm could be constructed.

Analysing the problem of false alarming, we found it to be a challenging problem to train a classifier to distinguish some of the false objects, such as the car-fans, some text areas on the car body, and the real car-plates. By observation, we found that car-plates are usually located at a fixed position and with a constant size. Therefore, searching in certain pre-defined regions, for a constant size, can effectively help avoid the false objects. For the localization of the main components in a car, we propose using a landmark localization method, previously used for face applications. This car-face landmark localizer will locate the main components of cars, especially the car-plate.

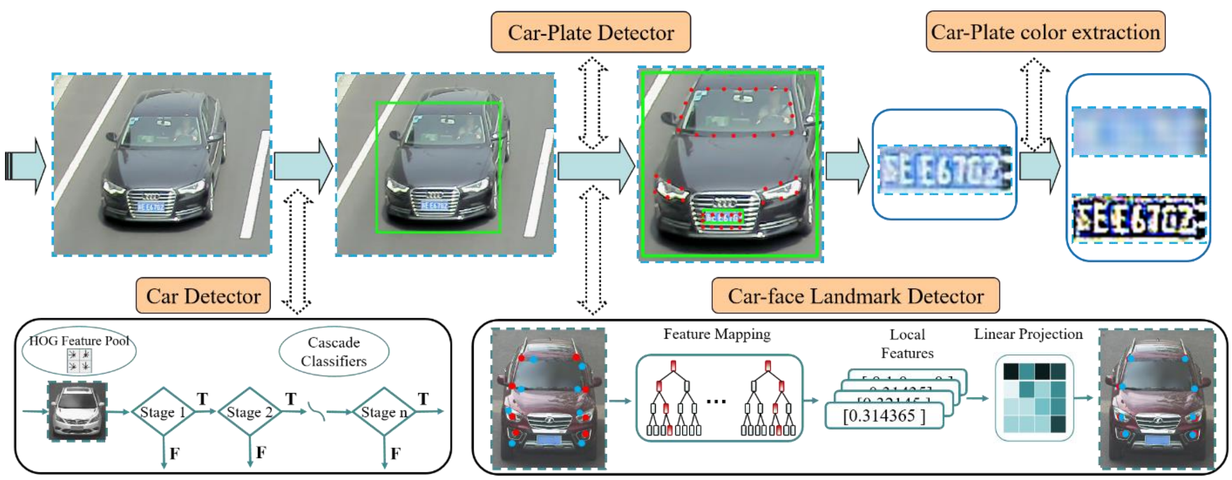


Figure. 3: Block diagram of our two-stage scheme for car-plate detection and car-plate color extraction

Aided by the landmark-localization method, which can better detect/track the targeted object, the most important components, i.e. the car-plate and the car nameplate areas, can be extracted more accurately. This can help the next car-recognition process become more accurate. In our application, we focus on the car-plate color extraction, which is a useful clue to detect car-plate areas more accurately. Although there are many variants of landmark-localization algorithms, for real-time applications, we have chosen a relatively faster algorithm proposed in [15]. Limited by its inherent capability, our training samples and the complex scene in out-door application, the car-face landmark localizer cannot extract the car-face landmark points accurately. However, we can set constrains for the car-plate detector, which considers both the size range of car-plates and their possible locations.

$$p = \max_{p_c \in \{D_{car-plate}\}, p_l \in \{L_{car-face}\}} \left\{ \| P_c \cap P_l \|^2_{overlapping} \; and \; \frac{\| P_c \cap P_l \|^2_{size}}{P_l} \right\} \tag{6}$$

which can be formulated as an unconstrained optimization problem as:

$$p = \max_{p_c \in \{D_{car-plate}\}, p_l \in \{L_{car-face}\}} \left\{ \| P_c \cap P_l \|^2_{overlapping} + \lambda \frac{\| P_c \cap P_l \|^2_{size}}{P_l} \right\} \tag{7}$$

where λ is a balance weight on approximated car-plate size and overlapping area.

As the car-plate detector, which is also trained from AdaBoost + HOG scheme, is able to extract the car-plate sensitively but with high false positive objects; Once the car-plate detector constrained by the car-face landmark localization, the effect of "two heads are better than one" works, which give out a performance with both high accuracy and low false alarming. As shown in Figure. 4, there are many candidates detected as car-plates in the three images. However, with our car-face landmark-localization algorithm, most of the false positives can be removed, while the true positives can be detected.

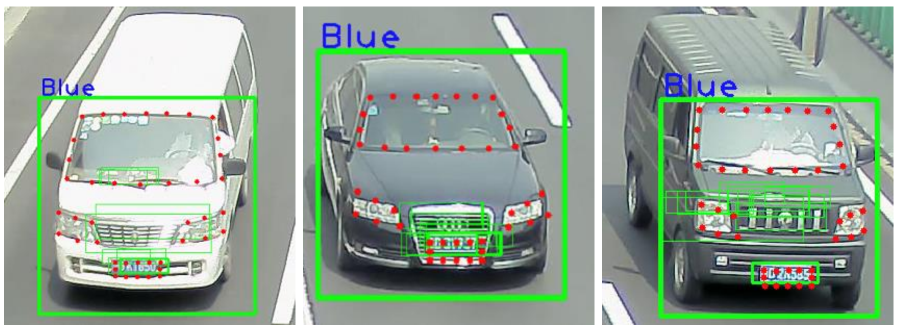


Figure. 4: Detection of car-plates from car-plate candidates.

The block diagram in Figure. 3 illustrates our proposed two-stage scheme for car-plate detection and car-plate color extraction. Once a blurred car-plate is extracted accurately, its color can then be extracted easily. We take the cartoon part as the color part and texture part as characters. We utilize the method from [13] to decompose the blurred car plate into different components, in which we can extract the car-plate color for later car type decisions by a trained SVM (Support Vector Machine) classifier. The whole algorithm is described in Table-1.

Two-stage scheme for car-plate detection and car-plate color extraction
1: Input and detectors used: a captured video frame (1920x1080 RGB) from a traffic surveillance system, the AdaBoost+HOG trained car detector D_{car}, and the AdaBoost+HOG trained car-plate detector $D_{car\text{-}plate}$, and a random-forest trained car-face landmark localizer $L_{car\text{-}face}$.
2: For each input image, the car detector D_{car} is employed to detect potential car candidates in the input image.
3: For each detected car candidate, the car-plate detector $D_{car\text{-}plate}$ is used to detect potential car-plates located in the bottom half of the car region. There is a high probability that the car-plate is located in that region.
4: For each detected car candidate, the car face landmark localizer $L_{car\text{-}face}$ is applied to extract the landmark points, in which the potential car-plate position can be obtained.
5: Compare all the car-plate candidates detected by the car-plate detector $D_{car\text{-}plate}$, the one which obtains the maximum value in (7) will be the best candidate.
6: Extract the blurred car-plate located in Step 5. The cartoon texture decomposition method is used to extract the cartoon part, which represents the color information about the car plate. Then, the RGB color histogram is used to determine its color type using a trained SVM classifier;
7: Output: the color type of an extracted blurred car-plate.

Table 1: The major steps for the two-stage scheme for car-plate detection and car-plate color extraction.

4. EXPERIMENTAL RESULTS

Color from a blurred car-plate can be extracted from its decomposed cartoon components [13]. The two examples in Figure. 5 show the results of the cartoon + texture decomposition algorithm. We can see that the texture component contains the information about character and numbers in the car-plate, although the texts are difficult to recognize. However, the color of the car plates can be extracted and detected accurately, which is useful information for subsequent car-type classification in some applications. For vehicle classification and traffic control for monitoring systems, most of the current literature aims only at detecting/tracking vehicles, without recognizing car-plate numbers. Given the different types of vehicles on the road (sedans, trail trucks, buses, etc.), counting all the vehicles and identifying their types will definitely be beneficial to the intelligent traffic surveillance and controlling systems. Furthermore, for counting vehicles in a highway scene, efficient trackers are used in our vehicle-detection project. Most of the time, the cars on a highway seldom cross lanes and usually maintain a distance from the vehicles in front and at back for safe driving (i.e. occlusion problem can be neglected in this situation). This allows us to use a relatively simple object tracker, such as the compressive tracker from [16], which can work efficiently and accurately with our application. Our proposed algorithm can work on images of resolution 1920×1080 pixels, with a speed higher than 30 frames per second, in an Intel i7 personal computer.

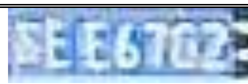
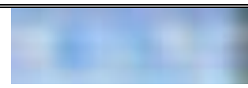
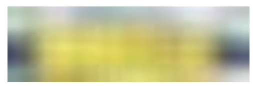
Samples	Original Car-plate	Cartoon Component	Texture Component
Blue car-plate			
Yellow car-plate			

Figure. 5: Color extraction with cartoon + Texture Image Decomposition.

5. CONCLUSIONS

Real-time on-road intelligent vehicle algorithms are still facing many challenging problems, even though many researchers have put their efforts into vehicle detection and tracking, and many algorithms have been proposed. A highly robust and reliable system is yet to be researched and developed. No single method can completely solve all the problems with intelligent traffic monitoring. Therefore, in this paper, we have proposed an efficient and accurate algorithm for detecting and locating blurred car-plates, and then use a decomposition algorithm to detect the colour information in a detected car-plate. The detected colour is useful for determining its vehicle type. The proposed approach can be a stepping stone to subsequent works on high-way surveillance systems.

REFERENCES

[1] H. A. Hegt, R. Haye, and N. A. Khan, A high performance license plate recognition system, IEEE Int. Conf. Systems, Man, Cybern., vol. 5, pp. 4357–4362, October 14, 1998.

[2] Y. Yanamura et al., Extraction and tracking of the license plate using hough transform and voted block matching, IEEE Int. Conf. Intelligent Vehicles Symp., pp. 243–246, June 9-11,2003.

[3] S.-Z. Wang and H.-J. Lee, Detection and recognition of license plate characters with different appearances, in Proc. of IEEE Int. Conf. Intelligent Transportation Systems, vol. 2, pp. 979–984 October 12-15, 2003.

[4] P. Xiang, Y. Xiuzi, and Z. Sanyuan, A hybrid method for robust car plate character recognition, in Proc. IEEE of Int. Conf. on Systems, Man, and Cybernetics, vol. 5, pp. 4733–4737, October 10-13, 2004.

[5] R. Zunino and S. Rovetta, Vector quantization for license plate location and image coding, IEEE Trans. Ind. Electron., vol. 47, no. 1, pp. 159–167, February, 2000.

[6] K. Kim, K. Jung, and H. Kim, Fast color texture-based object detection in images: Application to license plate localization, J. Lecture Notes Comput. Sci., vol. 177, pp. 297–320, 2005.

[7] P. Viola, and M. Jones, Robust Real-Time Face Detection, IJCV, vol. 57, no. 2, pp. 137-154, May, 2004.

[8] C. P. Papageorgiou and T. Poggio, A Trainable Object Detection System: Car Detection in Static Image, A.I. Memo No 1673, C.B.C.L Paper No 180. Massachusetts Institute of Technology, 1999.

[9] Y. Freund and R. E. Schapire, Experiments with a New Boosting Algorithm, in ICML, pp. 148-156, July 3-6, 1996.

[10] Md. Mostafa Kamal Sarker, Sook Yoon, Dong Sun Park: A Fast and Robust License Plate Detection Algorithm Based on Two-stage Cascade AdaBoost. TIIS 8(10): 3490-3507, 2014.
[11] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In CVPR, pages 886-893, 2005
[12] S. Ren, X. Cao, Y. Wei, J. Sun， Face Alignment at 3000 FPS via Regressing Local Binary Features, CVPR， 2014
[13] A. Buades, T. Le, J.M. Morel and L. Vese, Fast cartoon + texture image filters, TIP, 2010
[14] Timothy F. Cootes, Gareth J. Edwards and Christopher J. Taylor. Active Appearance Models, TPAMI, 2001
[15] Hailiang Li, Kin-Man Lam, Man-Yau Chiu, Kangheng Wu, Zhibin Lei. Cascaded Face Alignment via Intimacy Definition CoRR abs/1611.06642, (2016), https://ar-xiv.org/abs/1611.06642
[16] Kaihua Zhang, Lei Zhang, Ming-Hsuan Yang,Real-time compressive tracking, ECCV 2012.