IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Face Detection Mechanism Based on Machine Learning Techniques

Mr. Atul Kshitiz Sahu,
Student, Amity University Chhattisgarh

Mr. Ayushman Sengupta,
Student, Amity University Chhattisgarh

Dr. Mohammed Bakhtawar Ahmed,
Faculty, Amity University Chhattisgarh

Abstract

The multitude of applications of facial detection in ML has made it a topic of discussion. In this context, we have used methods based on machine learning that allows a machine to evolve through a learning process, and to perform tasks that are difficult or impossible to fill by more conventional algorithmic means. According to this context, we have established a comparative study between four methods (Haar-AdaBoost, LBP-AdaBoost, GF-SVM, GF-NN). These techniques vary according to the way in which they extract the data and the adopted learning algorithms. The first two methods "Haar-AdaBoost, LBP-AdaBoost" are based on the Boosting algorithm, which is used both for selection and for learning a strong classifier with a cascade classification. While the last two classification methods "GF-SVM, GF-NN" use the Gabor filter to extract the characteristics. From this study, we found that the detection time varies from one method to another. Indeed, the LBP-AdaBoost and Haar-AdaBoost methods are the fastest compared to others. But in terms of detection rate and false detection rate, the Haar-AdaBoost method remains the best of the four methods.

Keywords: Machine Learning, Haar-AdaBoost, LBP-AdaBoost, GF-SVM, GF-NN, Boosting, Cascade, Gabor.

I. Introduction

The objective of face detection is to find and locate faces in an image. It is the first step in automatic face recognition applications. Face detection has been well studied for frontal and near frontal faces. The Viola and Jones' face detector [1] is the most well-known face detection algorithm, which is based on Haar-like features and cascade AdaBoost [2] classifier.

Unconstrained scenes such as faces in a crowd, state-of-the-art face detectors fail to perform well due to large pose variations, illumination variations, occlusions, expression variations, out-of-focus blur, and low image resolution. numerous face detection methods have been developed following Viola and Jones', mainly focusing on extracting different types of features and developing different cascade structures.

Development in face detection has been to learn different cascade structures for Multiview face detection, such as parallel cascade [14], pyramid architecture [15], and Width-First-Search (WFS) tree [16]. All these methods need to learn one cascade classifier for each specific facial view (or view range).

In image pre-processing unit. Image normalization and illumination adaptation are some of the processes which is done on data in this module. Feature extraction module, is performed to provide effective information that is useful to distinguish faces and non-faces and stable with respect to the geometrical and photometrical variations. Finally, classification module is considered for classify face and non-face images based on the extracted features. In term of high accuracy and low false alarm, appearance-based methods which use learning machine algorithms as a class

In this paper we start with an introduction, then we present in the first section the detection methods based on Haar, LBP and Gabor extraction techniques. Then in the second section we expose the approaches of automatic learning Boosting, SVM and Neural Networks.

Then we present the results and the experiment containing a comparison between the four methods (Haar-AdaBoost, LBP-AdaBoost, GF-SVM) according to the processing time of the test images; the detection rate of the faces; and the rate of false detections.

II. FEATURE EXTRACTION

The pseudo-Haar features

Developed in 2001 by Paul Viola and Michael Jones, the Viola-Jones algorithm is an object-recognition framework that allows the detection of image features in real-time. Despite being an outdated framework, Viola-Jones is quite powerful, and its application has proven to be exceptionally notable in real-time face detection.

Only recently have our smartphones been able to use a human face as a password to unlock the device. Just like fingerprints, faces are unique with millions of tiny features that differentiate one from the other. It may not always be obvious to us humans, but machines synthesize and evaluate every small piece of data, leading to more objective accuracy.

Like other data-based models, Facial Detection is not 100% perfect. Although, it has reached a stage where it is commercially acceptable in our daily lives. Embedded in our devices, facial detection can be used in many ways, from simply unlocking your phone to sending money and accessing personal data.

There are two steps to the algorithms: there's training with facial and non-facial images and then there's the actual detection.

- We have 2 steps for training: training the classifiers and Ad boost
- We have 2 steps for detection: detecting the hair-like features and creating the integral image
- Viola-Jones is one of the most powerful algorithms of its time and even though, there are better models out there today, Viola-Jones set the foundation for it in the field of facial detection.

Local Binary Patterns (LBP)

LBPs is basically a texture descriptor made popular by Ojala et al. in their 2002 paper, although the concept of LBPs were introduced as early as 1993. The paper was based on Multiresolution Grayscale and Rotation Invariant Texture Classification with Local Binary Patterns.

LBPs compute a *local representation* of texture which is constructed by comparing each pixel with its surrounding neighbourhood of pixels.

The first step is to convert the image to grayscale. Then, we select a neighbourhood of size *r* surrounding the centre pixel in the grayscale image for each pixel. An LBP value is then calculated for this centre pixel and stored in the output 2D array with the same width and height as the input image.

Guided Filter SVM

We are aware with a method of HSI (Hyperspectral image) classification with SVM and guided filter. We extract the spatial features of HSI by the guided filter, which is obtained from the original HSI by a principal component analysis (PCA) method. We employ a guided filter again to optimize the classification.

Now, let's take a look on HSI (hyperspectral image)

Here hyperspectral imaging sensors have been widely used in remote sensing, biology, chemometrics, and so on. Hyperspectral image sensors can be obtaining spatial and spectral information of materials, due to abundant spectral information, HIS is widely applied to material recognition and classification, such as land cover, environmental protection, and agricultural. Hance HSI classification has attracted increasing attention and became a hot topic in the remote sensing community. The task of classification is to assign a unique label to each pixel vector of HSI. For this problem, many pixel-wise (spectral based) methods were employed, including k-nearest neighbours (KNN), support vector machine (SVM) and sparse representation.

Guided filter has been widely used in the field of noise reduction, image dehazing, and so no. we can get a new image that obtains feature of the guided filter.

III. CLASSIFICATION

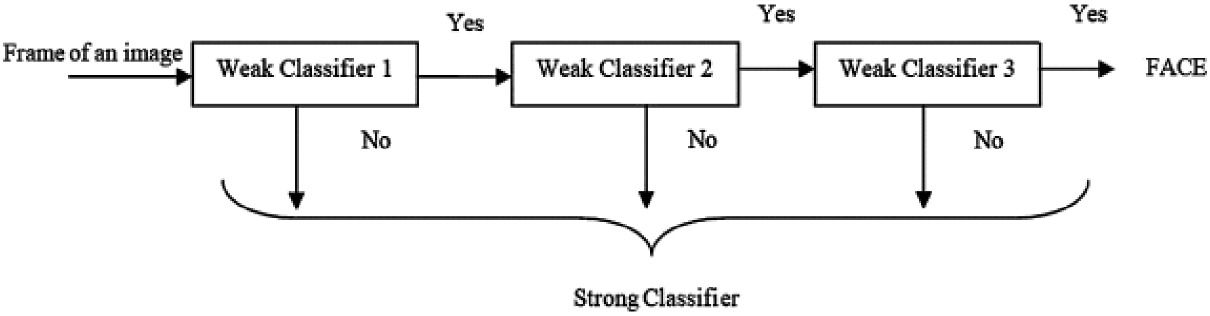
AdaBoost

1)Learning algorithm based on Adaboost:

An AdaBoost [1] classifier is a meta-estimator that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases. This class implements the algorithm known as AdaBoost-SAMME

2) Cascading classifiers:

Haar feature-based cascade classifiers is an effectual machine learning based approach, in which a cascade function is trained using a sample that contains a lot of positive and negative images. The outcome of AdaBoost classifier is that the strong classifiers are divided into stages to form cascade classifiers. The term "cascade" means that the classifier thus produced consists of a set of simpler classifiers which are applied to the region of interest until the selected object is discarded or passed.
The cascade classifier splits the classification work into two stages: training and detection. The training stage does the work of gathering the samples which can be classified as positive and negative. The cascade classifier employs some supporting functions to generate a training dataset and to evaluate the prominence of classifiers.
In order to train the cascade classifier, we need a set of positive and negative samples. In our work, we have incorporated the utility called opencv_createsamples to create the positive samples for opencv_traincascade. The output file of this function serves as an input to opencv_traincascade to train the detected face. The negative samples are collected from arbitrary images, which do not include the objects to be detected.



B) Support Vector Machine (SVM)

In machine learning, support vector machines are in control of learning models with associated learning models and algorithms that analyze data for classification and backward analysis.

A set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assign new examples to one category or the other, making it a non-probabilistic binary linear classifier. SVM map training examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

1) Linear classifier:

In addition to performing linear classification, SVMs can effectively perform a non-linear classification using what it is called a kernel trick, implicitly mapping their inputs into high-dimensional features spaces.
When the data are not labelled and separately/unsupervised learning is not possible, but its approach is required, which attempts to find natural assemble of data to groups, and then map/reconfigure new data to these formed groups. The support vector gather algorithm, applies the statistics of support vector, which was developed in the support vector machine algorithm, to categorized unlabeled data, and is one of the most widely used group of algorithms in industrial applications.

2 Maximum margin hyperplanes:

More formally a support vector machine constructs a set of hyperplanes in a infinite-dimensional spaces, which can be further used for classification, regression, or other tasks like outliers detection. A good separation is achieved by the hyperplane that has the largest distance to the nearest training data point of any also known as functional margin, since in general the larger the margin, the generalization error of the classifier. Our objective is to find a plane that has the maximum margin, that is the maximum distance between data points of both classes. it maximizes the margin distances provides some reinforcement so that the future data points can be classified with more confidence. Hyperplanes are decision boundaries that help classify the data points. Data points falling in either side of the hyperplane can be attributed to different classes. Also, the dimension of the hyperplane depends upon the number of features. If the number of input features is 2, then the hyperplane is just a line. If the number of input features is 3, then the hyperplane becomes a two-dimensional plane. It becomes difficult to imagine when the number of features exceeds 3. Support Vector are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. Using these support vectors, we maximize the margin of the classifier. Deleting the support vectors will change the position of the hyperplane. There are some points that helps us to build our SVM.

3) Multi-tasks classification on support vector machine

Learning multiple related tasks simultaneously has shown a better performance than learning these tasks in a separate way. Most approaches to multitask multiclass problems deteriorate them into multitask binary problems, and therefore it cannot effectively capture inherent correlations between classes. Although very elegant, traditional multitask support vector machines are restricted by the fact that different learning tasks must share the same set of classes.

Here we present a simple and short approach to multitask multiclass support vector machines based on the minimization of regularization functionals. We cast multitask multiclass problems into an artificial optimization problem with a quadratic objective function. Therefore, our approach can learn multitask multiclass problems directly and effectively. This approach can learn in two different scenarios: label-compatible and label-incompatible multitask learning. We can easily generalize the linear multitask learning method to the non-linear case using kernels. Several experiments, including comparisons with other multitask learning methods, indicate that our approach for multitask multiclass problems is very encouraging.

The most common aims of Multiclass Support Vector machine is to allot labels that are taken from a finite set of several elements to instances by using SVMs

The most dominant approach is to reduce the single multiclass problem into multiple binary classification problems. Common methods for such reduction include:

- Building binary classifiers that distinguish between one of the labels and the rest (*one-versus-all*) or between every pair of classes (*one-versus-one*). Classification of new instances for the one-versus-all case is done by a winner-takes-all strategy, in which the classifier with the highest-output function assigns the class (it is important that the output functions be calibrated to produce comparable scores). For the one-versus-one approach, classification is done by a max-wins voting strategy, in which every classifier assigns the instance to one of the two classes, then the vote for the assigned class is increased by one vote, and finally the class with the most votes determine the instance classification.
- Direct acyclic glass SVM (DAGSVM)
- Error code output codes.

4) Non-linear SVMs:

SVMs can effectively perform a non-linear classification using what it is called a kernel trick, implicitly mapping their inputs into high-dimensional features spaces.

5) Advantages:

The advantages of support vector machine are:

- Operative in high dimensional spaces.
- It is very implied in the situation where number of dimensions is much greater than the number of samples.
- It is very efficacious in memory where it uses a subset of training points in the decision function called support vectors, so it is also memory efficient.

- Versatile: it has ability to adapt different Kernel functions that can be specified for the decision function. Common kernels are available, but it is also possible to specify custom or modified stock Kernel.

6) Disadvantages:

The disadvantages of support vector machine:

- If the number of features/dimensions is much greater than the number of samples, avoid over-fitting in choosing Kernel functions and regularization term is crucial.
- These are calculated using an expensive five-fold cross -validation. Here probability is not directly provided by SVMs.

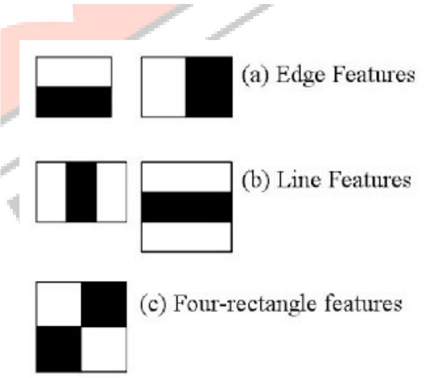
IV. EXPERIMENTAL RESULT

Haar-AdaBoost-:

Just look on Gentle AdaBoost algorithm which was used to train node classifier on a Haar-like features set to improve the generalization ability of the node classifier. Now a days, the face detection performance of the face detectors is improved.

Here we all know about that a computer only reads 0 and 1.

So, when the image is stored, it splits into 3 channels, that are composed of white light are Red, Green and Blue as we say 3 colour channels in a single pixel.



These channels are arrays with the same dimension. And the value stored in a cell of arrays represents the brightness of a colour channel in that pixel. These values are on a scale of 0 (no light) to 255 (the brightest).

Now, look in simple way that if a pixel is perfectly black, its value on the corresponding channels would be {R:0, G:0, B:0}. And if the pixel if perfectly white, their stored values on the arrays would be {R:255, G:255, B:255}

A Look Under the Hood

The Viola-Jones algorithm first begins with the change of image to grayscale. This makes the math easier, especially when you consider the nature of Haar-like features which are scalable, rectangular frames that are used to compare how pixels like the dark one relates the other.

There are three basic types of Haar-like features: Edge features, Line features, and Four-rectangle features. The white bars represent pixels that contain parts of an image that are closer to the light source and would therefore be “whiter” on a grayscale image. The black bars are the opposite. These are pixels whose image features are farther away from the light source (like a background) or are obstructed by another object (such as the eyebrows casting slight shadows over the eyes below). These features, like before, would appear “blacker” in a grayscale image. This comparison between white and black pixels is the most important reason that we transform the image to grayscale.

Important note: Remember that Haar-like features are *scalable*. In the case of Edge features, it could be 1x2, 100x200, or even 400x50 pixels. It doesn't matter! The only dimension they can't be is 1x1 pixels.

Edge features: These frames detect edges (simple enough). When it comes to face-detection, think of the forehead and the eyes/eyebrows. The forehead is an exposed, flat surface of the face. This allows it to reflect more light, so it is often "lighter". Eyebrows are generally darker. The algorithm would read the lighter shade of the forehead and the transition to the darker eyebrows as an "edge" boundary.

Line features: These detect? You guessed it! Lines. The pattern can go white-black-white, or black-white-black Going back to our example of face-detection, think about a nose. The top edge of your nose that stretches from the bridge to the nose tip, while not as flat as the forehead, is still reflective and the closest point on the face to a light source that might be in front of the subject, so it will naturally be brighter and stand out. The area around the nostrils typically bend away from the light making them darker. This pattern would be picked up as a line feature. Another interesting way that Line features are being utilized is in eye-tracking technology. Think about it: a **darker** iris sandwiched between the **white** space of your eye on either side of it. Pretty clever!

Four-Rectangle Features: This is good for finding diagonal lines and highlights in an image. This is used best on a micro scale. Depending on the lighting, it can pick out the edges of the jaw, chin, wrinkles, etc. These typically are features that aren't as important in general face-detection as there are so many of them, as well as so many variations in every individual's face, that it would lead to an algorithm that was too slow and might only detect the faces of certain people. In other words, too specialized.

As you can see, the algorithm classifies the transition from the forehead to the brow, the eyes to the cheeks, the upper-lip to the mouth, and the jaw to the chin as Edge-features. The nose follows a black-white-black pattern as light reflects off of the top. The highlight here creates a line and, thus, it is classified as a Line-feature.

A couple of notes should be made. First, this is just one example of how the algorithm *could* classify these parts of the face. Depending on the circumstances (whether or not the subject is wearing sunglasses), lighting (light source coming from a different angle), and scale (a group picture in front of the Taj Mahal, where facial features are a tiny part of a much bigger image) that the algorithm is working with, it could classify them differently.

Now, Haar-like features are defined by specific patterns of black and white pixels in a certain area. So the way in which the algorithm make decisions on a grayscale is quite simple that it explain in a Haar-like feature, if the difference between the means of the light and dark areas is within a certain threshold, treat them as black and white.

GF-SVM-:

It is a kind of edge -preserving smoothing filter. This image filter can also filter out noise or texture while retaining sharp edges. The guided image filter has two advantages: 1) it doesn't use too complicated mathematical calculations which has linear complexity. It is mathematically based on linear combination; the output image must be consistent with the gradient direction of the guidance image by which the problem of gradient reversal does not occur.

When using the bilateral filter an image, some artifacts may appear on the edges. This is because the pixel value abrupt changes on the edges. These artifacts are inherent and hard to avoid, because edges usually appear in all kinds of pictures. The guided filter performs better in avoiding gradient reversal. Moreover, in some cases, it can be ensured that gradient reversal does not occur.

V. CONCLUSION

In this paper we have summarized new methods for face detection systems based on four conventional machine learning algorithms. We have found that although detecting multiple frontal faces in images have been applicable but still detection faces in complex background with arbitrary viewpoint and different expression and occlusion need more attention. Also, some optimization method can be applied in face detection algorithm to find best features and reduce time of feature selection. Among existing methods for detecting faces in images, techniques using boosting algorithm are more effective for real time object detection.

REFERENCES

[1] Navabifar, Farhad, et al. "A short review paper on Face detection using Machine learning." *Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV)*. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2011.

[2] Shih, Peichung, and Chengjun Liu. "Face detection using discriminating feature analysis and support vector machine." *Pattern Recognition* 39.2 (2006): 260-276.

[3] Amaro, E. García, Marco Aurelio Nuño-Maganda, and Miguel Morales-Sandoval. "Evaluation of machine learning techniques for face detection and recognition." *CONIELECOMP 2012, 22nd International Conference on Electrical Communications and Computers*. IEEE, 2012.

[4] Filali, Hajar, et al. "Multiple face detection based on machine learning." *2018 International Conference on Intelligent Systems and Computer Vision (ISCV)*. IEEE, 2018.

[5] Sawat, Dattatray D., and Ravindra S. Hegadi. "Unconstrained face detection: a deep learning and machine learning combined approach." *CSI transactions on ICT* 5.2 (2017): 195-199.

[6] Abinaya, D., et al. "Classification of Facial Expression Recognition using Machine Learning Algorithms." *Journal of Physics: Conference Series*. Vol. 1937. No. 1. IOP Publishing, 2021.

[7] Abinaya, D., et al. "Classification of Facial Expression Recognition using Machine Learning Algorithms." *Journal of Physics: Conference Series*. Vol. 1937. No. 1. IOP Publishing, 2021.

[8] Crawford, Kate, and Trevor Paglen. "Excavating AI: The politics of images in machine learning training sets." *AI & SOCIETY* (2021): 1-12.

[9] Kim, Sanghyuk, Gwon Hwan An, and Suk-Ju Kang. "Facial expression recognition system using machine learning." *2017 international SoC design conference (ISOCC)*. IEEE, 2017.

[10] Damale, Radhika C., and Bazeshree V. Pathak. "Face recognition based attendance system using machine learning algorithms." *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*. IEEE, 2018.

[11] Muhmmad A U Khan, Muhmmad Khalid Khan, Muhammad Aurngzeb Khan, " Improved PCA based Face Recognition using Directional Filter Bank", IEEE International Multi topic Conference, pp 118-124, 2004.

[12]. Hala M. Ebied, " Feature Extraction using PCA and Kernel-PCA for Face Recognition", IEEE International Conference on Informatics and systems, pp 72-77, May 2012.

[13]. Patrik Kamencay, Martin Breznan, Dominik Jelsovka, Martina Zachariasova, " Improved Face Recognition Method based on Segmentation Algorithm using SIFT-PCA", IEEE International Conference On Tele communication and Signal Processing, pp 758-762, 2012.

[14]. Rammohan Mallipeddi and Minho Lee , " Ensemble Based Face Recognition using Discriminant PCA Features" IEEE International Conference On Computational Intelligence, pp 1-7, June 2012

[15] S. Bell, C. Lawrence Zitnick, K. Bala, and R. Girshick. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[16] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. In European Conference on Computer Vision, 2016.

[17] D. Chen, G. Hua, F. Wen, and J. Sun. Supervised transformer network for efficient face detection. In European Conference on Computer Vision, 2016.

[18] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. Intl. Journal of Computer Vision (IJCV), 2010.

[19] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg. Dssd: Deconvolutional single shot detector. arXiv preprint arXiv:1701.06659, 2017.