Proceedings of ICSP2020

Appearance-Based Gaze Tracking Through Supervised Machine Learning

Daniel Melesse, Mahmoud Khalil, Elias Kagabo, Taikang Ning, and Kevin Huang
Department of Engineering, Trinity College, Hartford Connecticut, USA
Email: {daniel.melesse, mahmood.khalil, elias.kagabo, taikang.ning, kevin.huang}@trincoll.edu

2020 15th IEEE International Conference on Signal Processing (ICSP) | 978-1-7281-4480-1/20/$31.00 ©2020 IEEE | DOI: 10.1109/ICSP48669.2020.9321075

***Abstract*—Applications that use human gaze have become increasingly more popular in the domain of human-computer interfaces, and advances in eye gaze tracking technology over the past few decades have led to the development of promising gaze estimation techniques. In this paper, a low-cost, in-house video camera-based gaze tracking system was developed, trained and evaluated. Seminal gaze detection methods constrained the application space to indoor conditions, and in most cases techniques required intrusive hardware. More modern gaze detection techniques try to eliminate the use of any additional hardware to reduce monetary cost as well as undue burden to the user, all the while maintaining accuracy of detection. In this work, image acquisition was achieved using a low-cost USB web camera mounted at a fixed position on the viewing screen or laptop. In order to determine the point of gaze, the Viola Jones face detection algorithm is used to extract facial features from the image frame. The gaze is then calculated using image processing techniques to extract gaze features, namely related to the image position of the pupil. Thousands of images are classified and labeled to form an in-house database. A multi-class Support Vector Machine (SVM) was trained and tested on this data set to distinguish point of gaze from input face image. Cross validation was used to train the model. Confusion matrices, accuracy, precision, and recall are used to evaluate the performance of the classification model. Evaluation of the proposed appearance-based technique using two different kernel functions is also assessed in detail.**

***Keywords* — human computer interface, automatic gaze tracking, support vector machine, face detection**

I. Introduction

Although there has been significant progress in the field of gaze tracking, commodity solutions are expensive and sometimes prohibitively so, thus limiting accessibility. To address this issue, this work integrates existing computer vision and classification algorithms to develop a low-cost appearance-based gaze tracking technique. The performance of the system is subsequently evaluated as a variety of gaze regions were analyzed. Classification of gaze coordinates was achieved using multi-class SVM, while K-fold cross validation verified robustness of classification with promising results. In total, up to nine separate gaze regions could be distinguished, while less granular divisions were also verified to work with the system. Two different kernel functions were used to assess the separability of the data in different states. After analyzing the performance of the technique, future improvements are suggested to increase accuracy.

A. Contributions

To the best of the authors' knowledge, the work proposed here contributes both

i) an appearance-based gaze tracking technique that uses only images acquired from a single web-camera;
ii) a performance analysis of the proposed technique using multiclass SVM for different kernel types.

The proposed approach aims to improve the field of gaze tracking. Results are promising, and have implications with regard to the possibility of achieving accurate gaze detection using non-intrusive commodity equipment, as shown in Fig. 1, for a large number of applications.

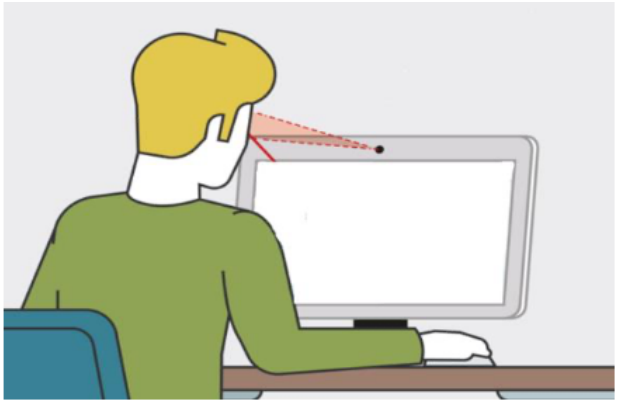


Fig. 1: Non intrusive and low-cost experimental gaze-tracking setup. Such a setup could expand the accessibility and prevalence of gaze-based human computer interfaces [1].

II. Background

Eye movement provides a rich and informative window into a person's intentions, and is used in day-to-day human-human interaction. It can give a direct indication of a user's point of attention, which opens another channel for human-computer interaction. Eye tracking can be used for several applications, including disease diagnosis, alertness level estimation, and activity recognition. During the past two decades, myriad of methods have been developed to estimate gaze movement. These can be broadly divided into two subclasses: intrusive and non-intrusive methods.

A. Intrusive Gaze-Tracking

While non-intrusive methods present a more streamlined and seamless approach to eye tracking, intrusive methods have benefits. One primary benefit of intrusive methods is increased accuracy. One intrusive technique involves electro-oculography, which computes the configuration of the eye by recording small differences in the skin potential around the region [2]. Another study used the principle of electromagnetism via devices called scleral search coils. Small coils of wire were inserted into a modified contact lens, which was subsequently applied to the human eye after local anesthesia. The contact lens allows tracing the coils orientation in a magnetic field, which in turn provides an estimate of the pupils' orientations and thus line of sight [3]. While these methods provide accurate results, they are often an uncomfortable burden and limit accessibility.

B. Non-Intrusive Gaze-Tracking

More recently, there is more focus on developing gaze tracking systems that are less invasive, thus making human-computer interaction more natural. Non-intrusive methods mostly use video-based eye tracking and can be roughly classified into two major categories: model-based and appearance-based methods. Model-based methods use a geometric eye model and can be further divided into corneal reflection and shape-based methods, depending on whether or not they require external light sources to detect eye features [4]. Early works on corneal reflection-based methods focused on stationary settings and were later extended to handle arbitrary head poses using multiple light sources or cameras. In contrast, shape-based methods directly infer gaze directions from observed eye shapes, such as pupil center or iris [5]–[8]. Although they have recently been applied to more practical application scenarios, these methods tend to result in lower accuracy, and it is unclear whether shape-based approaches can robustly handle low image quality and variable lighting conditions.

In contrast, appearance-based gaze estimation techniques extract input features from facial or eye appearance images and establishes a mapping relation to realize gaze estimation [9]. Common input eye features can be divided into three categories: complete human eye images, pixel-based features, and 3D reconstructed images [10]. Model-based gaze tracking methods require sophisticated hardware which may be composed by infrared light and high-definition cameras. In contrast, appearance-based methods usually only need a single camera to capture eye images. Certain eye features are generated from the complete eye images, and then a gaze mapping function is learned that maps eye image to gaze direction. Common eye features include a complete human eye image and the pixel-related information extracted from it. In order to perform such mapping, various regression techniques are used such as k-Nearest Neighbor (KNN), Random Forest, Support Vector Machines (SVM) and Artificial Neural Networks (ANN) [11]–[13].

SVM is found to be the most popular amongst regression techniques [14]. It solves linear separable problems first, however, in some cases it cannot find linearly separable partition planes. Therefore, SVM uses various kernels (radial basis and polynomial) to map the data into high dimensional space, and then assess the performance of each kernel. Thus, the performance of SVM depends on the kernel function and the selection of corresponding parameters [12].

Among gaze tracking systems that used SVM, Jian et al. developed a feature descriptor that feeds according to the location and scales of facial parts, which is then supplied to an SVM gaze classifier to obtain gaze direction [15]. Zhu et al. used pupil glint vector and 3D eye position to build an approximate generalized gaze mapping function [16]. On the other hand, Huang et al. used iris center as input features as the basis for the developed mapping function [7]. In general, SVM is robust, and the algorithm adjusts to the number of input features. An iris center can be sufficient to classify gaze direction [12].

C. Proposed Model

Most digital user environments exhibit platform with built-in web camera capabilities. This project utilizes this pervasive technology. The main challenge comes with the comparatively low image resolution and wide field of view – this means that only a small number of pixels are used to represent the eye making the determination of visual features difficult and error-prone [17].

Given an image of the user's face acquired from low-cost, built-in web cameras, the proposed method seeks to determine the location on the screen that the user is looking at. The approach uses support vector machine (SVM) on robust features to obtain reliable estimates of the eye position. The model is trained using cross validation based on extracted gaze coordinates. The model can then be used to make predictions of gaze coordinates given the extracted features of a new image, with granularity up to splitting the screen into 3×3, as depicted in Fig. 2. Confusion matrices, accuracy, precision, and recall metrics are used to evaluate the performance of the classification model.

Fig. 2: Nine classes of gaze regions classified with low-cost web cameras and the proposed SVM classification method. Lower resolution classification was also performed.

III. Methods

A. Image Acquisition

A C920 Logitech USB web camera was used for image acquisition of a single human test subject with minimal head movement. A standard LCD monitor was used, and a variety of gaze region granularities were tested, including splits of: 1×2, 2×2, 2×3, and 3×3. In total, 100 images were collected for each gaze region, i.e. 200 data images were collected for 1×2, 400 for 2×2 grid, 600 for 2×3 and 900 images for 3×3. This formed the database for this work. For each gaze region, 80 data points were designated for training and 20 data points for testing. In addition, face detection using the Viola Jones algorithm (detailed below) was empirically evaluated at different distances to determine optimal head positioning for image acquisition; a face-camera distance of 35cm was used.

B. Face and Eye detection

The Viola Jones algorithm is the first post-processing step in the method and is used to detect faces and facial features within an image. This algorithm uses Haar basis feature filters to detect facial features. Haar features are made up of a range of combinations of black and white pixels and are composed of either two, three, or four rectangles. Face candidates are scanned and searched for via Haar features of the current stage [18]. In practice, five primary patterns are considered, depicted in Fig. 3. The derived features are assumed to hold all the information needed to characterize a face.

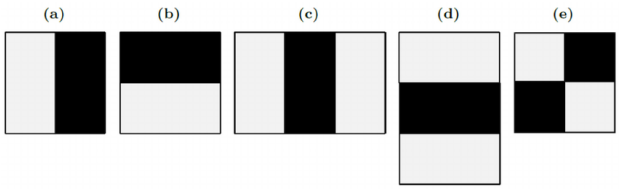


Fig. 3: Five primary Haar patterns used for the Viola Jones algorithm in this work.

1) Integral Image: To improve run-time efficiency, pre-computed integral images can be used to reduce Haar filter convolution complexity. Given pixel located at coordinate (x, y) in an integral image, the pixel value $p(x, y)$ is defined as

$$p\,(x,\ y) = \sum_{i \le x, j \le y} I(i', j') \tag{1}$$

Where, $p(x, y)$ the integral image pixel value and $I(x, y)$ is the pixel value at (x, y) in the original image. The integral image is essentially the double integral of the image. The integral image reduces computational time when convolving the Haar patterns with the original image, which can then be calculated by adding/subtracting only four numbers.

In this work, the Viola Jones algorithm initiates with a 24×24 kernel for Haar feature detection. Considering all permutations of position, scale and type results in over 160,000 possible features. Most of these are non-essential for face detection and can be eliminated via AdaBoost training.

2) AdaBoost Training: AdaBoost selects required features to train the data set to enhance performance of the classifier function. The procedure effectively identifies a select set of good "features" (e.g. eye and mouth features). A weighted combination of these features which perform better than chance form a set of weak classifiers. Usually, AdaBoost reduces the number of features to about 2,500. Linear combinations of weak classifiers form strong classifiers.

3) Cascading Classifier: The Viola Jones algorithm final procedure consists of a cascade of stages. The overwhelming majority of sub-regions in an image are not faces, and that most of these sub-regions can be identified as non-faces with little computation. Thus, the first stage of the detector contains a computationally efficient classifier to eliminate non-faces. Only a small fraction of sub-regions are then passed onto the next stage for further processing. This process repeats with higher complexity non-face identification, thus the average computational effort per sub-region remains low. The results of interest for this method are the segmented eye regions from facial images. This pipeline is summarized in Fig. 4

Fig. 4: Viola Jones detects both face and eye region.

C. Pupil Detection

Once eye regions are extracted via the Viola Jones algorithm, a Hough circle transform is used to detect the radius and the center coordinates of the pupil, which then can be used to determine gaze direction. The main advantage of the Hough transform technique is that it is tolerant of gaps in feature boundary descriptions and remains relatively unaffected by image noise, unlike edge detectors [19], [20]. A sample extracted pupil center is calculated from the left eye in Fig. 4. The result is shown in Fig. 5.

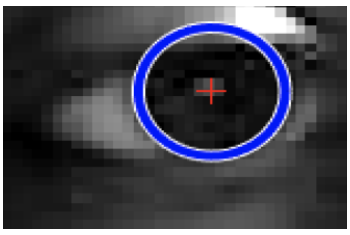
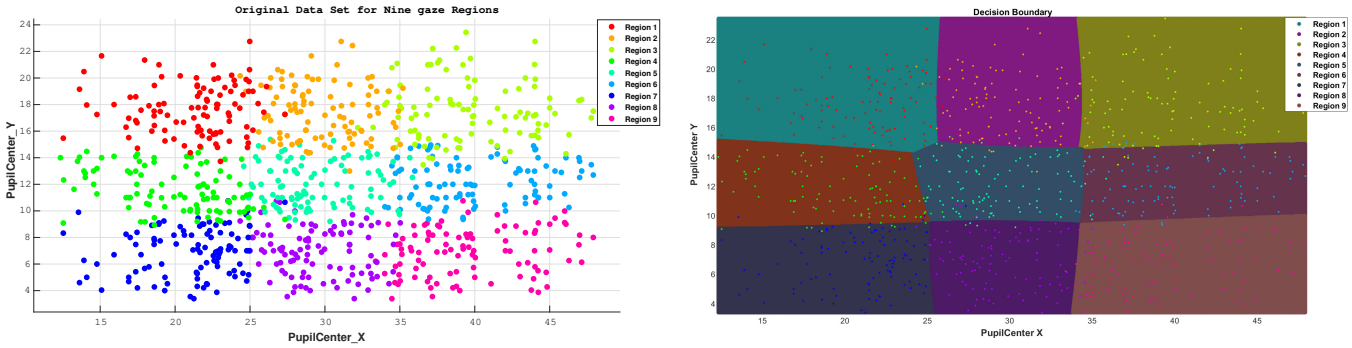


Fig. 5: Example pupil center detection using Hough circle transform.



(a) Original data set for 3×3 grid.

(b) Decision Boundary for 3×3 grid using quadratic kernel

Fig. 6: The original 3×3 data set contains 900 pupil center data points, i.e. 100 points per gaze region. The nine separate different gaze regions are then classified based on the quadratic kernel based SVM classifier, with final decision boundaries defined in (b). Different granularity gaze region divison and the RBF kernel are also used for comparison.

D. Support Vector Machine

Pupil center coordinate feature vectors were grouped and labeled based on gaze region. An SVM-based classifier was trained to map pupil coordinates into gaze regions. Supervised SVM simultaneously minimizes the empirical classification error and maximize the geometric margin by finding the hyperplane that maximizes the margin between two classes.

1) Kernel Function: The collected gaze data is non-linear separable. While linear classification of non-linear data can be achieved in a higher dimension, determining such a transformation can be computationally expensive. Instead, a non-linear kernel can be used to reduce the computational complexity [21]–[23]. In this work, two such kernels are tested, namely a polynomial kernel and radial basis function (RBF) kernel.

The polynomial kernel is defined as

$$K(x, y) = (x^T y + c)^d$$

where in this work, c was chosen as 1, and d as 2. This second order polynomial kernel is thus quadratic [24].

The radial basis function kernel is represented mathematically as

$$K(x, y) = \exp(-\gamma ||x - y||^2)$$

where γ is the inverse of training data set cardinality for each grid type. γ is also used as a similarity measure between two data points. Intuitively, a small γ value indicates an RBF with a large variance; the model is too constrained and cannot capture the complexity of the data [24].

2) Multi-class SVM: SVM is fundamentally a binary classifier, which can be extended to multi-class problems. One-against-all classification was chosen for its simplicity; one binary SVM is trained for each class to separate members of that class from members of other classes. To train and validate the SVM model, cross validation was performed. 80% of the gaze data was used as a training data set to fit the SVM model while 20% of the gaze data was used as a testing data, selected uniformly across all classes.

IV. Results

Figure 6 shows the labeled training dataset for 3 × 3 resolution and quadratic kernel decision boundaries overlaid. The 3×3 case is the most complex of the resolutions tested. To evaluate performance of the two kernels, confusion matrices were calculated for the 3 × 3 case, shown in Fig. 7

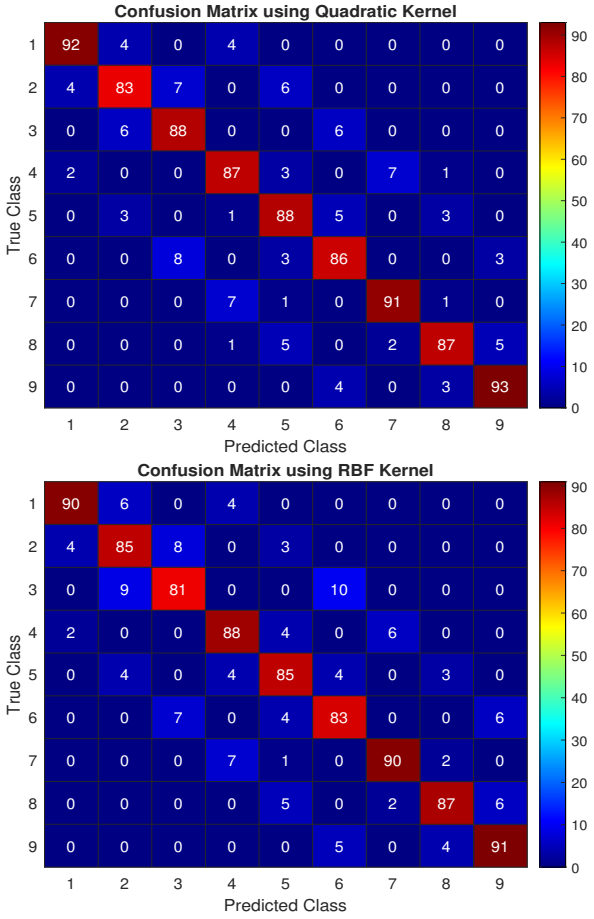


Fig. 7: Confusion matrices for 3×3 resolution case using quadratic (top) and RBF (bottom) kernels.

To compare the two kernels in more detail, accuracy of classification was compared for each across all four gaze region resolutions. These scores were calculated and are tabulated below in Table I.

TABLE I: Kernel accuracy across gaze region resolutions.

Grid Type	Kernel	Accuracy
1×2	Quadratic	96.7
	RBF	95.7
2×2	Quadratic	94.5
	RBF	93.3
2×3	Quadratic	89.5
	RBF	89
3×3	Quadratic	88
	RBF	85.3

From the above analysis, the quadratic kernel performed better than RBF. Precision, recall and subsequent F_1 scores were then calculated to evaluate the quadratic kernel SVM classifier for each class. These results are shown in Table II

TABLE II: Precision, recall and F_1 score

Gaze Region	Precision	Recall	F_1 score
Region 1	93.8	92	92.9
Region 2	86.5	83	84.7
Region 3	85.4	88	86.7
Region 4	87	87	87
Region 5	83	88	85.2
Region 6	85. 1	86	85.6
Region 7	91	91	91
Region 8	91.6	87	89.2
Region 9	92	93	92.5

V. Conclusion

In this work, a low-cost, multiclass supervised learning gaze tracking technique was prototyped and tested. Cross validation was used to train the model, two nonlinear kernels (quadratic and RBF) were tested. Performance metrics of the classifiers using confusion matrices as well as calculated accuracy show that the quadratic kernel outperformed the RBF kernel. The use of web cameras dictates that the image resolution will be lower. Since the user's eyes take up a small fraction of the captured image, the actual image size that we work with is on the order of 60×30 pixels for the whole eye. With minimal head movement, the method was still able to distinguish nine different gaze regions.

Future work involves the training and testing of the method using live streaming video. Time complexity must be performed to accurately guarantee on-line performance and determine if a pre-processing stage needs to be optimized or replaced. Utilizing artificial neural networks (ANNs) instead of support vector machine may also provide better performance as higher resolution classification is desired.

Acknowledgements

This work was supported by the Trinity College Engineering senior capstone design program. The authors would like to thank Drs. John Mertens and Harry Blaise, as well as Andrew Musulin for technical support and guidance.

References

[1] "Online eye-tracking per webcam," Jul 2019. [Online]. Available: https://innofact-marktforschung.de/en/methods/online-eye-tracking/

[2] M. P.A.Punde and R.R.Manza, "A study of eye tracking technology and its applications," *1st International Conference on Intelligent Systems and Information Management*, no. 6, pp. 86–90, 2017.

[3] J. W. Bang, E. C. Lee, and K. R. Park, "New computer interface combining gaze tracking and brainwave measurements," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 4, 2011.

[4] J. Y. R. Stiefelhagen and A. Waibel, "A model-based gaze tracking system," *IEEE International Joint Symposia on Intelligence and System*, no. 3, pp. 304–310, 1996.

[5] A. B. L. Swirski, "Robust real-time pupil tracking in highly off-axis images," *Proceedings of the Symposium on Eye Tracking Research and Applications*, pp. 173–176, 2012.

[6] E. S. J. Wang and R. Venkateswarlu, "Eye gaze estimation from a single image of one eye," *Computer Vision Proceeding*, 2003.

[7] K. Huang, S. Petkovsek, B. Poudel, and T. Ning, "A human-computer interface design using automatic gaze tracking," in *2012 IEEE 11th International Conference on Signal Processing*, vol. 3, Oct 2012, pp. 1633–1636.

[8] K. Huang, M. Khalil, E. Luciani, D. Melesse, and T. Ning, "A data-driven approach for gaze tracking," in *2018 14th IEEE International Conference on Signal Processing (ICSP)*. IEEE, 2018, pp. 494–499.

[9] D.Gong and K.Kwak, "Face detection and status analysis algorithms in day and night environments," *International Conference on Advanced Informatics, Concepts, Theory, and Applications*, no. 8, pp. 1–4, 2017.

[10] K. Wang and Q. Ji, "Real time eye gaze tracking with 3d deformable eye-face model," *IEEE International Conference on Computer Vision*, no. 10, pp. 1003–1011, 2017.

[11] S. Baluja and D. Pomerleau, "Non-intrusive gaze tracking using artificial neural networks," *Advances in Neural Information Processing Systems*, vol. 98, no. 1, p. 753–760, 1993.

[12] Y. C. W. H. Wu, Y.L., "Gaze direction estimation using support vector machine with active appearance model," p. 2037–2062, 2014.

[13] S. J. B. M. A. E. S. Palmero, C., "Recurrent cnn for 3d gaze estimation using appearance and shape cues," *Computer Vision Pattern Recognition*, vol. 1, no. 3, pp. 1–13, 2018.

[14] Y. Zhang, "Support vector machine classification algorithm and its application," *International Conference on Information Computing and Applications*, no. 7, pp. 179–186, 2012.

[15] C. Jian-nan, Z. Chuang, Q. Yan-jun, L. Ying, and Y. Li, "Pupil tracking method based on particle filtering in gaze tracking system," *International Journal of Physical Sciences*, vol. 6, no. 5, pp. 1233–1243, 2011.

[16] Z. Zhu, Q. Ji, and K. P. Bennett, "Nonlinear eye gaze mapping function estimation via support vector regression," in *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 1. IEEE, 2006, pp. 1132–1135.

[17] X. et al, "Interested object detection based on gaze using low-cost remote eye tracker," *International IEEE/EMBS Conference on Neural Engineering*, no. 9, pp. 1101–1104, 2019.

[18] M. J. J. P. Viola, "Robust real-time face detection," *International journal of computer vision*, vol. 57, no. 2, pp. 137–154, 2004.

[19] I. D. M. Smereka, "Circular object detection using a modified hough transform," *International Journal of Applied Mathematics and Computer Science*, vol. 18, no. 1, pp. 85–91, 2008.

[20] J. K. J. Illingworth, "A survey of the hough transform," *Computer vision, graphics, and image processing*, vol. 44, no. 1, pp. 87–116, 1998.

[21] Y.-Y. O. C.-Y. C. Z.-W. C. Yen-Jen Oyang, Shien-Ching Hwang, "A novel learning algorithm for data classification with radial basis function networks," *Proceedings of the 9th International Conference on Neural Information Processing*, 2002.

[22] D. S. C. Arti Patle, "Svm kernel functions for classification," *International Conference on Advances in Technology and Engineering (ICATE)*, 2013.

[23] H. X. Zhiliang Liu, "Kernel parameter selection for support vector machine classification," *Journal of Algorithms & Computational Technology*, vol. 8, no. 2, 2013.

[24] G. Liu, J. Yang, Y. Hao, and Y. Zhang, "Effect of different kernels on the performance of an svm based classification," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 7, 2019.