Pedestrian Group Detection with K-Means and DBSCAN Clustering Methods

Mingzuoyang Chen, Shadi Banitaan, Mina Maleki, Yichun Li
Dept. of Electrical & Computer Engineering & Computer Science
University of Detroit Mercy
Detroit, MI, United States
chenmi7@udmercy.edu, banitash@udmercy.edu, malekimi@udmercy.edu, liyi8@udmercy.edu

2022 IEEE International Conference on Electro Information Technology (eIT) | 978-1-6654-8009-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/eIT53891.2022.9813918

***Abstract*—The development of autonomous vehicles has made real-time pedestrian detection and tracking an important research area for protecting human lives and improving society. A key challenge in this area is to improve pedestrian detection accuracy while reducing the processing time for tracking. This challenge can be addressed by detecting and tracking pedestrian groups since pedestrians move in groups. This paper proposes a method of detecting pedestrian groups using unsupervised machine learning methods. After detecting the pedestrians, K-Means and DBSCAN clustering methods were used to detect pedestrian groups based on the coordinates of pedestrians' bounding boxes. Simulation results of the MOT17 dataset indicate that both clustering techniques could reduce detection and tracking processing times. However, K-Means clustering is more effective than DBSCAN clustering in detecting pedestrian groups as measured by the Silhouette Coefficient score and Adjusted Rand Index (ARI).**

***Index Terms*—DBSCAN, K-Means, Object detection, Object tracking, Group detection**

I. Introduction

As autonomous driving technology develops and becomes more prolific, engineers have started to focus on pedestrian tracking since it can not only help with the traffic but also protect pedestrian safety. It commonly uses the locations that are provided by pedestrian detection. Currently, there are several parameters that can be used to locate a pedestrian on the street, such as using points [1]–[3] and bounding boxes [4]–[6], each having their own strengths.

Due to the complicity of the environment, pedestrian tracking is a complex task. For the vehicles on the road, they need to take suitable reactions based on the pedestrian direction. If we can track the pedestrians, it will be helpful to prevent accident occurrences [7], since the driver may not be able to pay attention to their view on the road at all times. As a result, we may need to use technologies like object tracking to alert drivers to these conditions, potentially reducing the number of accidents.

In most of the traditional methods for pedestrian tracking, they generate an ID number for each of the pedestrians they detect in object detection. When working on the blocked pedestrian situation, the method needs to reconnect the missing human to its previous trajectory or create a new ID for them once they are re-detected. Because of this, the methods might take some time. When working in a heavy crowd, tracking these pedestrians will take even more time than expected. In order to reduce the processing time, both object detection and object tracking need to improve.

Since there are already so many well-developed object detection algorithms that operate with reasonable accuracy, improving the real-time detection and tracking processing times has recently attracted attention [8]–[10]. When we are on a street, we frequently observe individuals walking in groups, which causing a new idea called group detection emerged. It separates the objects into different groups, reducing the overall number of tracking objects, which can deal with both missing object and processing time. Since there are few papers working on clustering the pedestrians, this paper mainly focuses on pedestrian group detection. We will compare different group detection methods, K-Means and DBSCAN, to determine the most effective one.

The following is the outline of the paper. We describe the purpose of our study in Section I. Section II introduces the methods for pedestrian tracking and group detection, followed by an explanation of the methods in Section III. In section IV, we compare different approach results and discussions. The conclusion and future work will be discussed in the final part.

II. Related work

Object detection is a computer technology based on computer vision and machine learning that allows us to locate specific objects in the image or the video [11]. It is always the most time-consuming step in pedestrian trajectory prediction. Most of the state-of-the-art (SOTA) object detection methods have great accuracy, but it requires many trials to get these results [12], [13]. Since we are using it in a vehicle, both speed and accuracy are important, which means these methods might not be suitable for the situation we need.

Nowadays, when we look at the road, we notice pedestrians walking in groups. They might be with couples, families, or friends, which means they might walk at same the speed and have the same direction [8], [14], [15]. Even for strangers with different speeds, they might keep in the same group with their neighbor for several frames, which results in the idea of the group detection.

Previous works on group detection using pedestrian trajectories to classify the groups, like [16], develop a Multiview-based Parameter-Free framework, it first detects the objects according to the points, then run the feature extraction in

978-1-6654-8009-3/22/$31.00 ©2022 IEEE

order to find the motion of objects, and at last use the information to do the clustering base on the tightness. In [17], the authors introduce the sociologically inspired features to extend the previous works, they have the same concept on the intuition of the pedestrian groups, so they defined a series of features to extract the peculiarities of groups. The G-MITRE precision and recall shows that their thoughts work. [8] used DBSCAN clustering methods, as they are trying to introduce the coexisting time ratio as another input to the DBSCAN method and they compared it to other group detection methods which apply coexisting time or distance base on the interaction over union. The result shows that their proposed method is more accurate and reliable.

According to these methods, group search can be seen as a potential direction for pedestrian detection. When working on the pedestrian detection, if we apply group detection, it will highly reduce the calculation time on the following steps, such as object tracking, since we lower the object number. But all these methods are working on an overhead camera, which is not suitable for us because we want human view and the zooming rate for a overhead camera is different from human view.

III. Material and Methods

Group detection can be formulated as a clustering task. There exist several well-known methods that can be used for clustering, such as K-Means, DBSCAN, Agglomerative Clustering, etc.. The goal of this research is to apply group detection to the object detection result. The overall framework is described in Fig. 1. The dataset we are using is MOT17-02 [18], which provides a fixed camera of human view video with 600 frames, it records the pedestrians moving in a street for about 20 seconds. We first forward the image to object detection method, then filter the result with object filtering. We apply different kinds of group detection methods to the filtered results, and track grouped objects we get from the previous step and compare the processing time.

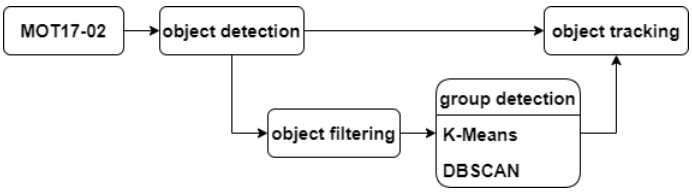

Fig. 1. Structure of pedestrian trajectory prediction

A. Object detection

Fig. 2 shows the original image with the result of object detection. The method we used is the given model on pytorch [19], which is fasterrcnn_resnet50_fpn, it is composed of a Faster R-CNN model, with the 50 layers ResNet backbone [20], combined with Feature Pyramid Networks(FPN). R-CNN, which is a region-based convolutional neural network, is a class of deep neural networks. It uses selective search to choose regions from the image, then runs the SVM to classify the output and get the correct labels. Fast R-CNN, unlike R-CNN, feeds the region proposals to the CNN, then uses the ROI pooling layer to reshape the image into a fixed size to let it be fed into the network. The reason why Faster R-CNN is used by most of the SOTA methods is because of the region proposal network (RPN), which is the backbone of the method. It uses this separate network instead of selective search to make the network learn the region proposals by itself.

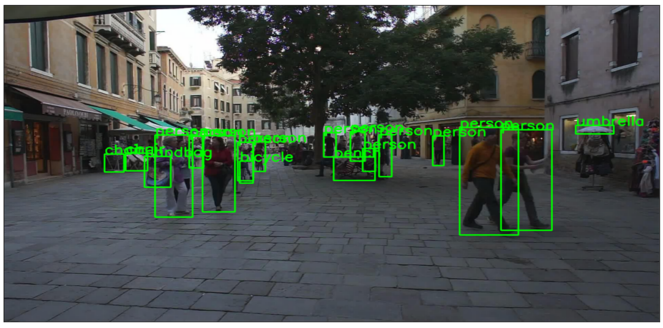

Fig. 2. Original object detection result

B. Object Filtering

Different types of pictures and videos can be used as inputs. Our approach is mainly based on .jpg images, and the dataset we utilized in our methods is the MOT17-02, which contains images with a 1920*1080 image size. We will apply the object detection algorithm straight to this dimension without any down sampling, and then collect the result of all the objects inside. The predicted label and the coordinates of the bounding boxes are returned by the model we employ for object detection. The bounding boxes are represented by the top-left and lower-right points.

In our method, we filter away non-pedestrian objects initially since we are attempting to work on the pedestrian. Then we will group the items that are not difficult to detect. When performing object detection on an image, we usually notice that people who are far away from the camera are constantly absent, despite the fact that they are not blocked by other items. This happens because the camera is set to human vision, and each row of the image has a different zooming effect. Pedestrians who are far away from the camera will shrink in size, making object detection methods difficult to detect. We need to define a threshold to filter out these things that appear inconsistently so we do not waste time on them.

Because the zooming rate is different in the image, the bounding boxes of the long-distance pedestrians are smaller, but the size cannot be the parameter to measure the distance between the camera and pedestrians. When a child crosses the street, the small size of the bounding box could correspond to a much closer distance to the camera, so we filter the pedestrians based on the lower-right point on the bounding boxes. Whether the human is in large size or small size, the point on their feet will show exactly where they are. As a result, the threshold is set at half of the image length, and all items above it are

eliminated. Shown in Fig. 3 is the sample image being filtered by the label and threshold.

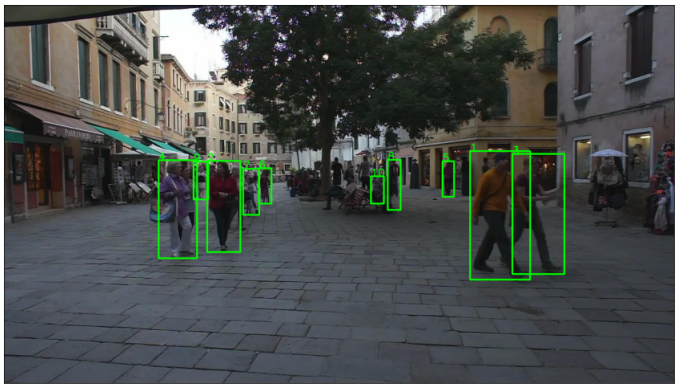

Fig. 3. Object detection result with filtering

C. Group detection

In this paper, we apply two kinds of group detection methods, which are both cluster-based algorithms, to the object detection results.

K-Means clustering [21] is a method for grouping data points into k clusters based on their distance from the nearest mean. The approach will randomly select k beginning points, compute the Euclidean distance between each point and these initial points, and cluster the points based on these values, then recalculate the means and set the new value as the cluster points. This process is repeated until the cluster points do not change. This approach is a quite efficient way to cluster data, but we must specify the k value manually, and the methods will stack on the local minimum, causing the clustering result to be less reliable than expected.

DBSCAN [22] stands for Density-Based Spatial Clustering of Applications with Noise, and is a density-based method that can cluster the points based on the distance and number of neighbors. A cluster is a dense region of points, which is separated by low-density regions, from other regions of high density. The inputs of this method are epsilon, which is the number of points within a specified radius, and minimum points nearby(MinPts). Unlike K-Means clustering method, it did not need to set the number of clusters. The algorithm first compares the distance between each other, and if there exists two points distance smaller than the epsilon, it will set one point as the neighbor of the other point. The points have not less than the MinPts neighbor, it will be called core point, otherwise, if it has neighbors that less than MinPts, it will be called border point, if neither situation, then it will be called noise point. We did not specify the MinPts since we are working with pedestrians and we cannot assume the pedestrian is an error. Since the camera is in human view, which means we cannot cluster the pedestrians only based on the information we got from object detection, we set dynamic epsilon based on the sorted distance to the nearest point of each object. We calculate the slope from the smallest distance to the largest distance as the threshold, and then compute the change in neighboring value from the minimum to the maximum. We then compare it to the threshold, and once it is larger, we set the smaller value of compared distance as the epsilon to cluster the pedestrians.

D. Object tracking

Object tracking is a method for continually locating an object in a video. It requires the information from the given frame in order to acquire information from the same objects in subsequent frames. The methods cannot detect and track an object on their own, they can require object detection to obtain information such as object coordinates, or they can use other sorts of codes, such as cv2.selectROI in Python, to allow users to select the object that need to be tracked.

After many years of development, several object tracking methods exist that can have high accuracy while dealing with various types of distractions. Because of its speed, DCF-based tracking algorithms (discriminative correlation filters) are widely utilized. Such as the Kernels correlation filter, which uses multi-channel features to increase accuracy, and the CSR-DCF tracking algorithm, which combines CNN features to improve robustness [23]. There are various open-source object tracking algorithms available online, including BOOSTING, MIL, KCF, TLD, MEDIANFLOW, GOTURN, MOSSE, and CSRT, which are all included in OpenCV.

IV. Results and Discussion

The purpose of this research, as indicated earlier, is to find a suitable clustering method for pedestrian grouping. To accomplish this, we employ the Silhouette Coefficient score and ARI metric as an evaluation metric to discover the appropriate pedestrian grouping methods, we also utilize the sum-of-squared errors (SSE) to select the number of groups in K-Means method. After that, we tested each method numerous times to find if our hypothesis, that group detection can speed up object tracking processing, was correct. Our research is now focused on grouping the pedestrians and comparing total processing time with and without the group detection.

To the best of our knowledge, there is no dataset which contains the ground truth on group detection, hence the experiment was conducted to obtain the group labels for each pedestrian, which aims to evaluate the accuracy of clustering by using ARI. In this experiment, we have only two annotators and we gave them both the original video and the videos for each pedestrians with unique ID. After watching the videos several times, they generate the group list for all the pedestrians in ground truth except for those who are far away from camera. We test the inter-annotator agreement by cohen's kappa score, which measures the agreement between annotators on the clustering result. It can be calculate by equation 1, where p_0 is the empirical probability that a given sample is correctly labeled, and p_e is the expected agreement between two annotators when they randomly assign labels. A cohen's kappa score of 0.9282 has been obtained

by sklearn.metrics.cohen_kappa_score, which indicates a high matching rate between two annotators.

$$K = \frac{(p_0 - p_e)}{(1 - p_e)} \tag{1}$$

A. Evaluation metrics

The SSE is a standard metric for the K-Means clustering method. The error here refers to the distance between the points and their nearest cluster, and the SSE is the sum of these values squared. The value will decline once the points are clustered to the right group and the k value can be set at the "elbow" of the K-SSE curve [24].

The Silhouette Coefficient score is commonly used to evaluate the quality of the clustering methods. It needs to calculate two values, which is the mean distance from the points to all the other points that are in the same group denoted as 'a' and the mean distance from the points to all the other points in its next nearest cluster denoted as 'b'. The equation to calculate the Silhouette Coefficient is the equation 2 below. The value of it is between -1 and 1, the larger the better, when it becomes a negative value, it means the points are being clustered into a wrong group.

$$S = \frac{(b - a)}{max(a, b)} \tag{2}$$

The ARI function measures the similarity between the predicted clusters and group truth clusters. The equation of ARI is defined as equation 3, where RI represents Rand Index, which takes all points identified within the same cluster into consideration to discover if two clusters are similar [25]. The output is guaranteed to be closed to 0 for random labeling, and exactly 1 when identical clusters are presented.

$$ARI = \frac{(RI - expected(RI))}{(max(RI) - expected(RI))} \tag{3}$$

B. Analysis of the clustering methods

In this paper, we employ K-Means and DBSCAN clustering methods on the pedestrian detection results. We all know that the number of clusters k is the input of the K-Means methods; any incorrect k will result in unsuitable clustering. After filtering the object detection result, we first apply K-Means methods with various k values. In our experiments, we use the k values from 4 to 8 since there only exist 10 pedestrians in the first frame of our video, to compute the SSE, and use the value of "elbow" as the number of clusters in our methods. The SSE plot is presented in Fig. 4, the elbow of the plot can be used as the optimal value of k. As can be observed, k = 6 is the appropriate number for our methods.

The group detection result using K-Means approach for a sample image is shown in Fig. 5, where we found that the pedestrians are grouped as we expected. However, when using K-Means clustering method in the pedestrian grouping, we must choose k value in each detection period, and the window size of the k value selected is difficult to specify. When pedestrians are few or in large crowds, we specify the

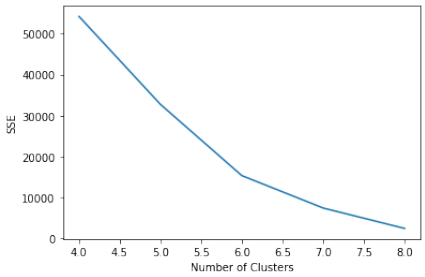

Fig. 4. SSE of different clusters

k value in different types, like changing the range of k and the value distance. Also the optimal value of k for now we can only discover by observation, since the SSE plot might make the k value chosen stuck at a local minimal value.

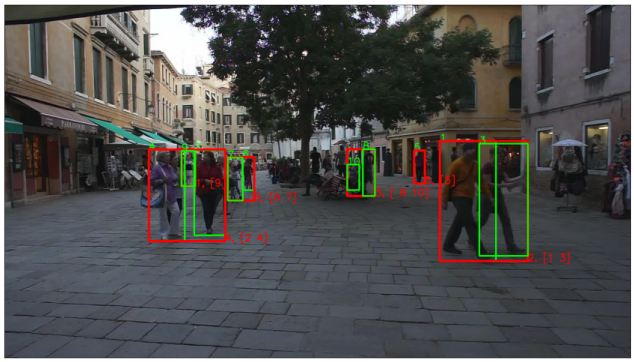

Fig. 5. Group detection with K-Means method (k=6)

DBSCAN clustering algorithms, on the other hand, are density-based and no need to care about the number of clusters. In our testing, we did not set the MinPts, and chose epsilon based on the slope of the curve. As we mentioned earlier, we use NearestNeighbors in sklearn to calculate the distance to its nearest neighbor for setting the epsilon. Fig. 6 shows the grouping result with DBSCAN methods and we can observe that the clustering result is not perfect. This happened because of the zooming rate of the image.

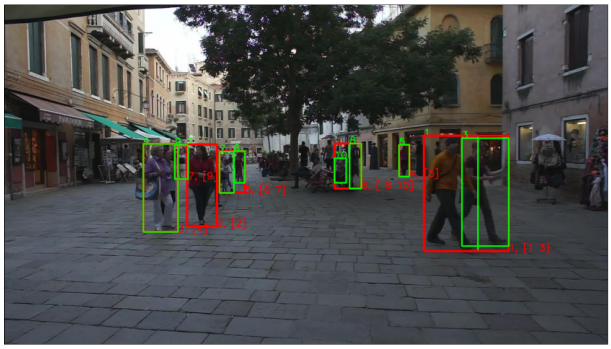

Fig. 6. Group detection with DBSCAN method

We first evaluate the results by the model itself. In our code,

we evaluate our results based on Silhouette Coefficient score, we used a given evaluation function in sklearn [26], which is sklearn.metrics.silhouette_score, it is used to compute the score of all the points inside the matrix. The inputs of it contain the matrix of bounding box coordinates and the predict labels and output the evaluation result. Table I shows the Silhouette Coefficient score of each kind of methods. The scores show that the K-Means method outperforms the DBSCAN method.

We also apply ARI function to evaluate our model. We apply another model from sklearn, which called sklearn.metrics.adjusted_rand_score. It require predicted labels and true labels as the input, based on the experiment we create, we set up the ground truth by ourselves. Table I shows the comparison on Silhouette Coefficient score and ARI metrics, which shows that K-Means method had the perfect clustering, and DBSCAN method also got a good result. According to these scores, we found that K-Means method have more accurate score than DBSCAN method, but the k value chosen of K-Means make the clustering become more complex, which cause the DBSCAN method more acceptable in my opinion.

TABLE I
COMPARISON ON DIFFERENT GROUP DETECTION METHODS

Methods	K-Means	DBSCAN
Silhouette Coefficient	0.460	0.416
Adjusted Rand Index	1.000	0.845

C. Analysis of the effect of group detection on object tracking

Object tracking is the main part of our evaluation, because our goal is to reduce the processing time on object tracking. In this experiment, we used the CSRT tracker included in OpenCV package to track pedestrians or groups of pedestrians (the group bounding box coordinates) obtained from group detection model, because this tracker is fast and robust to disturbances. Each method is tested five times, and the average value of the total processing times, which includes both detection and tracking parts, on 600 frames is used as the tracking time. In table II, we show the comparison of different method results. As shown in the table, while the K-Means method takes a bit longer than DBSCAN, both methods achieve the goal of reducing processing time of object tracking.

TABLE II
COMPARISON OF THE TOTAL PROCESSING TIMES

Round(sec.)	Original	K-Means	DBSCAN
Average	555.52	529.87	514.10

V. CONCLUSION

In this work, we proposed a framework to detect pedestrian groups to reduce processing time for tracking. The first step is to detect pedesterians in each frame. Next, clustering techniques including K-Means and DBSCAN were used to identify pedestrians walking as a group. The simulation results on the MOT17-02 dataset demonstrates the feasibility of using group detection to reduce object tracking processing time. The proposed approach lacks consideration for different zooming rates in each frame, which is one of its limitations. In the future, we will try to solve this problem by improving clustering methods. We will also use other datasets to test the efficiency of the proposed method using a variety of camera viewpoints.

REFERENCES

[1] Mínguez, R. Q., Alonso, I. P., Fernández-Llorca, D., & Sotelo, M. A. (2018). Pedestrian path, pose, and intention prediction through gaussian process dynamical models and pedestrian activity recognition. IEEE Transactions on Intelligent Transportation Systems, 20(5), 1803-1814.

[2] Ludl, D., Gulde, T., & Curio, C. (2020). Enhancing data-driven algorithms for human pose estimation and action recognition through simulation. IEEE transactions on intelligent transportation systems, 21(9), 3990-3999.

[3] Abughalieh, K. M., & Alawneh, S. G. (2020). Predicting pedestrian intention to cross the road. IEEE Access, 8, 72558-72569.

[4] Zhang, Y., Wang, C., Wang, X., Zeng, W., & Liu, W. (2021). Fairmot: On the fairness of detection and re-identification in multiple object tracking. International Journal of Computer Vision, 1-19.

[5] Karthik, S., Prabhu, A., & Gandhi, V. (2020). Simple unsupervised multi-object tracking. arXiv preprint arXiv:2006.02609.

[6] Han, S., Huang, P., Wang, H., Yu, E., Liu, D., Pan, X., & Zhao, J. (2020). Mat: Motion-aware multi-object tracking. arXiv preprint arXiv:2009.04794.

[7] Haddad, S., Wu, M., Wei, H., & Lam, S. K. (2019). Situation-aware pedestrian trajectory prediction with spatio-temporal attention model. arXiv preprint arXiv:1902.05437.

[8] Cheng, H., Li, Y., & Sester, M. (2019, June). Pedestrian group detection in shared space. In 2019 IEEE Intelligent Vehicles Symposium (IV) (pp. 1707-1714). IEEE.

[9] Wang, R., Cui, Y., Song, X., Chen, K., & Fang, H. (2021). Multi-information-based convolutional neural network with attention mechanism for pedestrian trajectory prediction. Image and Vision Computing, 107, 104110.

[10] Zhu, J., Chen, S., Tu, W., & Sun, K. (2019). Tracking and simulating pedestrian movements at intersections using unmanned aerial vehicles. Remote Sensing, 11(8), 925.

[11] Dasiopoulou, S., Mezaris, V., Kompatsiaris, I., Papastathis, V. K., & Strintzis, M. G. (2005). Knowledge-assisted semantic video object detection. IEEE Transactions on Circuits and Systems for Video Technology, 15(10), 1210-1224.

[12] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28.

[13] Dai, J., Li, Y., He, K., & Sun, J. (2016). R-fcn: Object detection via region-based fully convolutional networks. Advances in neural information processing systems, 29.

[14] Sawas, A., Abuolaim, A., Afifi, M., & Papagelis, M. (2018, June). Tensor methods for group pattern discovery of pedestrian trajectories. In 2018 19th IEEE International Conference on Mobile Data Management (MDM) (pp. 76-85). IEEE.

[15] Zaki, M. H., & Sayed, T. (2017). Automated analysis of pedestrian group behavior in urban settings. IEEE Transactions on Intelligent Transportation Systems, 19(6), 1880-1889.

[16] Li, X., Chen, M., Nie, F., & Wang, Q. (2017, February). A multiview-based parameter free framework for group detection. In Thirty-First AAAI Conference on Artificial Intelligence.

[17] Solera, F., Calderara, S., & Cucchiara, R. (2015). Socially constrained structural learning for groups detection in crowd. IEEE transactions on pattern analysis and machine intelligence, 38(5), 995-1008.

[18] A. Milan, L. Leal-Taix´e, I. Reid, S. Roth, and K. Schindler, "Mot16: A benchmark for multi-object tracking," arXiv preprint arXiv:1603.00831, 2016.

[19] Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., ... & Lerer, A. (2017). Automatic differentiation in pytorch. Available: https://pytorch.org/vision/stable/models.html

[20] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[21] MacQueen, J. (1967, June). Some methods for classification and analysis of multivariate observations. In Proceedings of the fifth Berkeley symposium on mathematical statistics and probability (Vol. 1, No. 14, pp. 281-297).

[22] Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996, August). A density-based algorithm for discovering clusters in large spatial databases with noise. In kdd (Vol. 96, No. 34, pp. 226-231).

[23] Liu, S., Liu, D., Srivastava, G., Połap, D., & Woźniak, M. (2021). Overview and methods of correlation filter algorithms in object tracking. Complex & Intelligent Systems, 7(4), 1895-1917.

[24] Yuan, C., & Yang, H. (2019). Research on K-value selection method of K-means clustering algorithm. J, 2(2), 226-235.

[25] Sinnott, R. O., Duan, H., & Sun, Y. (2016). Chapter 15-a case study in big data analytics: exploring twitter sentiment analysis and the weather. Big Data, 357-388.

[26] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011