ISSN 2224-087X. Електроніка та інформаційні технології. 2024. Випуск 25. С. 81–92
Electronics and information technologies. 2024. Issue 25. P. 81–92

UDC 004.932 DOI: https://doi.org/10.30970/eli.25.7

COMPARISON OF SUPPORT VECTOR CLASSIFICATION WITH KEY POINTS AND NEURAL NETWORKS FOR OBJECT DETECTION

V.-T. Luchka, V. Vdovychenko, Y. Furgala

Ivan Franko National University of Lviv,
50 Drahomanova St., 79005 Lviv, Ukraine
VasylTaras.luchka@lnu.edu.ua

Object detection plays a crucial role in computer vision applications, ranging from autonomous driving to facial recognition. Over the years, researchers have developed various techniques to tackle the challenges of object detection. Among them, feature detection algorithms and neural networks have emerged as powerful approaches. This paper aims to provide a comparative analysis of these two methodologies, exploring their strengths and weaknesses.

Key words: Key points, support vector classification, neural networks, SIFT, ORB, BRISK, FREAK, YOLOv5

Introduction. The task of object detection is widely recognized and challenging within the industry, and it has been significantly addressed through the advancements in computer vision (CV) utilizing machine learning (ML) techniques. But robustness and efficiency of feature detection algorithms still makes them a viable solution for special tasks. Feature detection algorithms find utility in applications where real-time performance, interpretability, and limited computational resources are key, such as robotics and embedded systems. On the other hand, neural networks excel in scenarios where accuracy and the ability to handle complex visual patterns are essential, including autonomous driving, surveillance, and large-scale object detection tasks.

The research investigates various methods for detecting and describing key points in images and their application in machine learning for image recognition. The algorithms under consideration include SIFT [1], ORB [2], BRISK [3], and FREAK [4], along with a custom feature-based object detection function utilizing key point coordinates, the SVC [5-7] method with image descriptors, and the YOLOv5 neural network.

The study aims to compare the speed of key point detection and descriptor computation among the different algorithms using a specific dataset. Average times will be calculated to evaluate the performance of each algorithm based on the number of key points detected.

Building upon the comparative analysis of the algorithms, an attempt will be made to develop a novel object detection function solely based on key point coordinates. The performance of this function will be assessed to determine its effectiveness.

In addition, a well-established method described in external research [8], the SVC method, will be employed for object classification using key point descriptors. An SVC model will be created, trained on the provided dataset, and its performance and computational metrics will be evaluated in comparison to speed and accuracy performance of YOLOv5 neural network. SVM belongs to supervised learning paradigm of machine learning, so comparisons between SVM and neural networks are feasible.

To provide a comprehensive comparison with algorithmic object detection methods, an alternative approach utilizing neural networks will be employed. Models of different sizes will be trained and assessed for their speed and performance in object recognition tasks.

By conducting these comparisons and evaluations, the research aims to gain insights into the efficiency and quality of the various methods for object detection and recognition, shedding light on their potential applications and performance in real-world scenarios.

Input data collection and preparation. For this research custom made dataset was created. This dataset consists from 153 images of 4 different soda cans brands. Pictures were made from different angels and different distance.

Fig. 1. Dataset composition

For speed comparison of feature detection algorithms the same 140 images form dataset were used. I was recording next metrics:

1. Number of key points detected.
2. Time to detect key points.
3. Time to compute descriptors

Fig. 2. Example of key point detection with SIFT

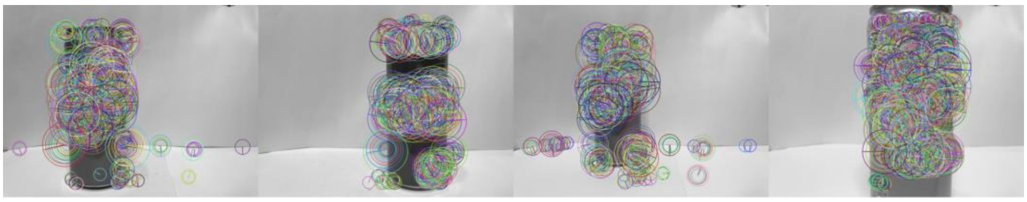

Fig. 3. Example of key point detection with ORB

Table 1. Computational speed comparison of SIFT, ORB, BRISK and FREAK

Algorithm	Average number of key points detected	Average time of key points detection (ms)	Average time of descriptor computation (ms)
SIFT	231.32	67	46
ORB	1123.1	3	4
BIRSK	420.8	7	4
FREAK (SIFT)	220.2	72	28

Based on data obtained during tests, we can claim that Oriented FAST and Rotated BRIEF (ORB) [2, 9] and Binary Robust invariant scalable keypoints (BRISK) [3] algorithms are the fastest in key points detection and descriptor calculation. But how good are they for object detection? We will 2 methods for object detection using key points. First it's self written solution function based on fingerprint of distances from key points, other one using Support Vector Classification (SVC).

Fingerprint method consist of next steps:

1. Match of key points between template image (our "perfect" image of object) and image where we are trying to detect object
2. Calculate matrix of distances between key points on template image and image with object we are trying to detect
3. Calculate crosscorrelation of matrixes.
4. Calculate sum for each crosscorrelation matrix.
5. Assume that object belongs to class with biggest correlation sum.

Using fingerprint method gave us insufficient results.

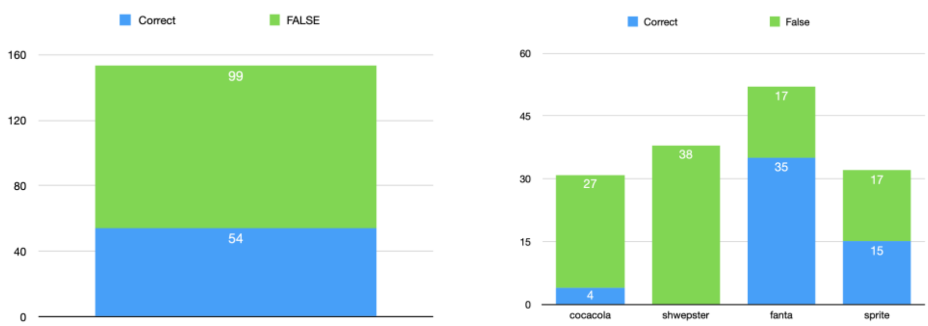

Fig. 4. Results of fingerprint method.

Such bad results of fingerprint method could be contributed to handcrafted design: the performance of this method highly depends on the quality and selection of handcrafted features, which can be challenging and time-consuming. There is room for improvements but if it is worth it, this question could not be answered in scope of this work.

Support Vector Classification (SVC) is a widely used algorithm in machine learning that belongs to the family of Support Vector Machines (SVMs). SVC is particularly effective for binary classification tasks, where it seeks to find an optimal hyperplane that separates data points belonging to different classes. In this essay, we will explore the fundamental concepts of SVC, its key features, advantages, and limitations, as well as its applications in various domains.

SVC aims to find a decision boundary that maximally separates data points of different classes in a feature space. It achieves this by mapping the original data into a higher-dimensional space using a kernel function, which enables the identification of a hyperplane that effectively separates the classes. SVC finds the hyperplane by maximizing the margin, which is the distance between the hyperplane and the closest data points of each class.

To use SVC for object detection with key points we will train Linear SVM on dataset and then use calculated model to detect objects on unseen previously images.

For training Linear SVM model we will perform next steps:

1. Calculate key points and descriptors from our dataset (some pictures are excluded from training dataset, to verify final accuracy with images that were not used in training)
2. Stack all the descriptors vertically in a numpy array
3. Perform k-means clustering
4. Perform Tf-Idf vectorisation
5. Perform scaling of our features
6. Train the Linear SVM

To use trained model for object detection we need to perform next steps:

1. Load pertained model
2. Calculate key points and descriptors form image
3. Stack all the descriptors vertically in a numpy array
4. Perform Tf-Idf vectorization
5. Perform the predictions
6. Report predicted class

Fig. 5. Results of predictions using SVC model with SIFT, ORB and BRISK key points and descriptors

Fig. 6. Example of wrong prediction with SVC model

Table 2. Comparison of time needed for training and testing of SVC models on different key point detection algorithms and resulted accuracy

Algorithm of key point detector and descriptor calculation	Time to execute train program (s)	Time to fit the model (ms)	Predicted accuracy	Accuracy based on 20 test images	Time to execute test program (s)	Time to perform detection on test images (ms)
SIFT	35	5	100%	100%	6.0	0.16
ORB	46	10	100%	100%	3.4	0.25
BRISK	45	10	100%	95%	3.4	0.18
FREAK (SIFT)	32	10	100%	90%	5.1.	0.25
FREAK (ORB)	17	12	100%	100%	3.3	0.33
FREAK (BRISK)	29	12	100%	100%	3.6	0.15

To test SVC models for robustness test images with obstructions, different lighting conditions and different resolution were used.

ISSN 2224-087X. Electronics and information technologies. 2024. Issue 25

Fig. 7. Result of SVC model predictions with SIFT, ORB and BRISK key points and descriptors (from left to right)

Fig. 8. Result of SVC model predictions with FREAK-SIFT, FREAK-ORB and FREAK-BRISK key points and descriptors (from left to right)

Table 3. Comparison of time needed for predictions with SVC and resulted accuracy

Algorithm of key point detector and descriptor calculation	Time to execute prediction program (s)	Time to predict (ms)	Accuracy
SIFT	35.5	5.7	75%
ORB	6.5	0.15	25%
BRISK	7.9	0.37	50%
FREAK (SIFT)	31.0	2.15	25%
FREAK (ORB)	6.5	0.15	0%
FREAK (BRISK)	8.2	0.34	50%

Neural networks. For neural network YOLOv5 (You Only Look Once version 5) was chosen. YOLOv5 is an advanced object detection framework that builds upon the success of its predecessors to deliver state-of-the-art performance with efficient architecture, Single-stage Detection, Anchor-Free Localization and Improved Backbone Network. While YOLOv5 may have limitations in small object detection and contextual understanding, its strengths in speed, accuracy, and real-time inference make it a compelling choice for various computer vision applications, including autonomous driving, surveillance, and robotics. With ongoing research and improvements, YOLOv5 continues to push the boundaries of object detection, driving advancements in the field of computer vision.[10, 11]

I used 3 models with different number of epochs of training:

1. Model S with 32 epochs.
2. Model S with with 128 epochs.
3. Model M with 32 epochs.

All models were trained on VPS server form Linode with 4 cores and 8 Gb of RAM.

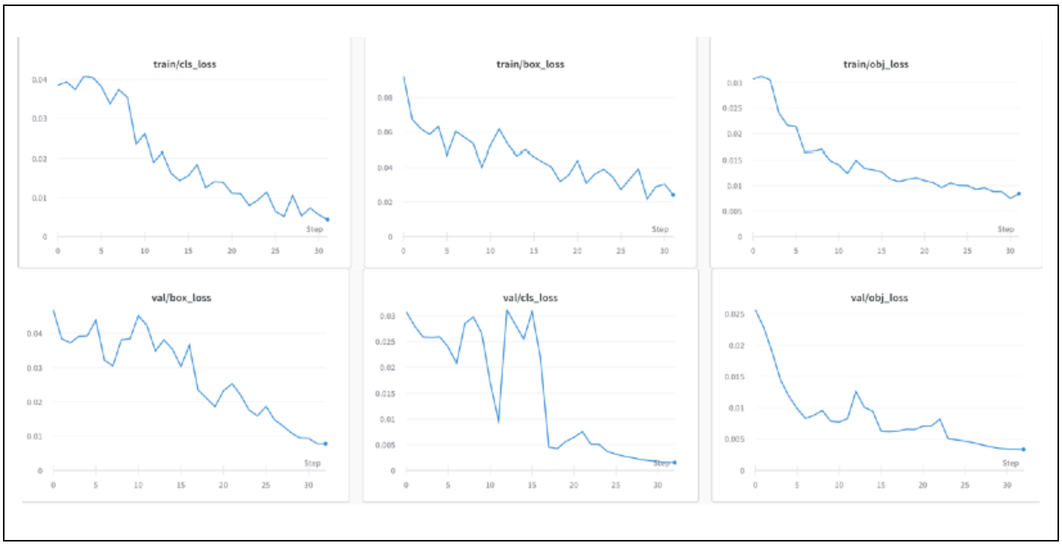

Fig. 9. Train and validation loss of YOLO v5 model S with 32 epochs

Even on this small and fast to train model (0.442 hours) we can see very respective results for object detection.

Fig. 10. Results of model S with 32 epochs

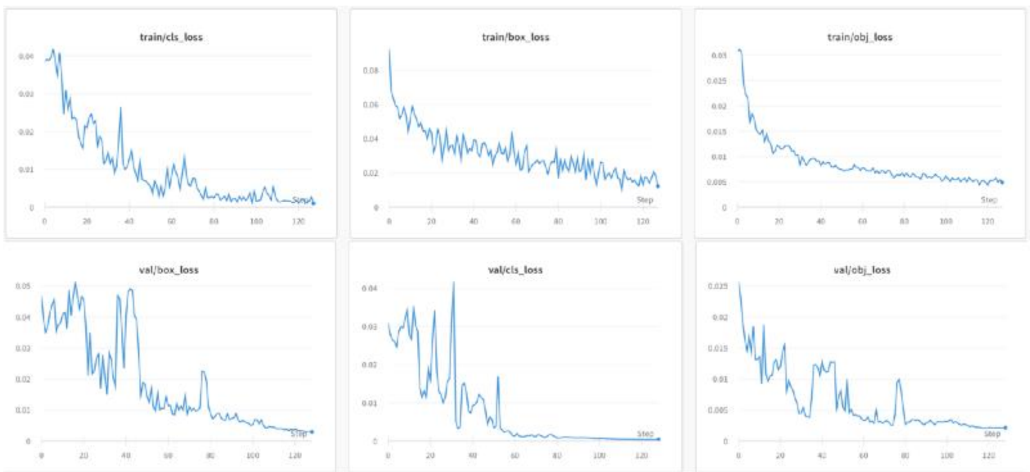

Fig. 11. Train and validation loss of YOLO v5 model S with 128 epochs

Fig. 12. Results of model S with 128 epochs

In reference to results of model S with 128 epochs and it’s train and validation losses we can see wast redundancy in training and possible overtraining of neural network.

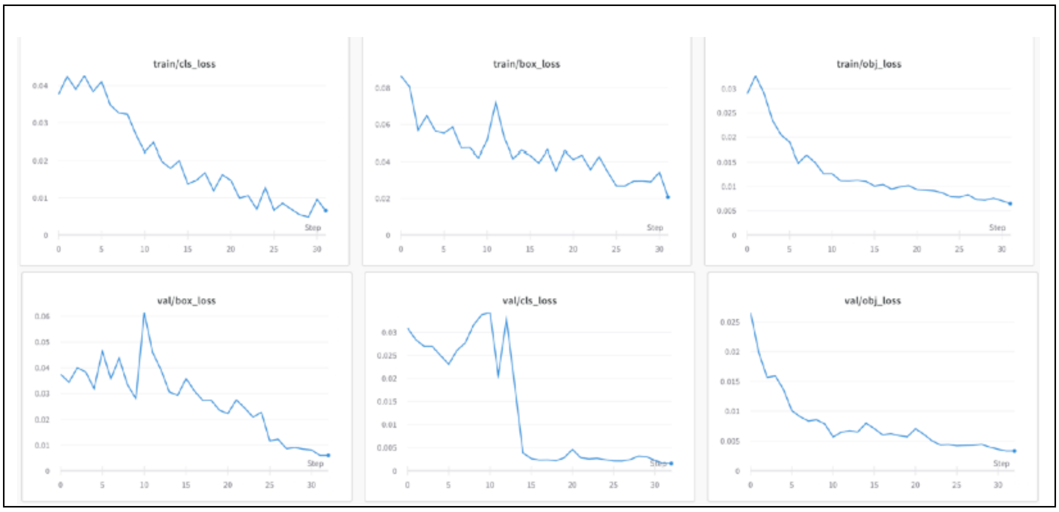

Fig. 13. Train and validation loss of YOLO v5 model M with 32 epochs

Fig. 14. Results of model M with 32 epochs

Table 4. Comparison of time needed for training and prediction for different YOLOv5 models

Model size	Number of epochs	Train time (h)	Average time for detection (ms)
S	32	0.442	89
S	128	1.621	93
M	32	8.783	246

Conclusion. Comparison of feature detection algorithms and neural networks is tough. It's two completely different fields of computer science. In our case, when key points are used as a crucial part of object detection program which relies on Support vector machine (SVM) we can compare its performance to neural networks. Only fair metrics to compare are accuracy and speed. In this case, in training category we have clear winner, Support vector classifier in conjunction with key paints from one of algorithms (SIFT, ORB, BRISK) is at least 45 times faster then training of YOLOv5 network. Average time for detection also is smaller in SVC, but this difference is not so significant. For real-time applications this could make a huge difference, especially in embedded solutions where computational power is limited. Both methods (SVC and neural networks) have excellent accuracy, when tested on previously unseen images. YOLOv5 should perform better in case it sees image of different scale because of its convolutional layers and other operations to extract semantic information from the image at various scales and levels of abstraction. Time for detection with limited computational power is not great, but could be easily improved this graphics accelerators.

REFERENCES

[1] *Lowe, D. G.* (2004). Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60(2), 91-110.
[2] *Rublee, E., Rabaud, V., Konolige, K., & Bradski, G.* (2011). ORB: An efficient alternative to SIFT or SURF. In Proceedings of the IEEE International Conference on Computer Vision (ICCV) (pp. 2564-2571). IEEE.
[3] *Leutenegger, S., Chli, M., & Siegwart, R. Y.* (2011). BRISK: Binary robust invariant scalable keypoints. In Proceedings of the IEEE International Conference on Computer Vision (ICCV) (pp. 2548-2555). IEEE.
[4] *Alahi, A., Ortiz, R., & Vandergheynst, P.* (2012). FREAK: Fast retina keypoint. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 510-517). IEEE.
[5] *Cortes, C., & Vapnik, V.* (1995). Support-vector networks. Machine learning, 20(3), 273-297.
[6] *Burges, C. J.* (1998). A tutorial on support vector machines for pattern recognition. Data mining and knowledge discovery, 2(2), 121-167.
[7] *Hsu, C. W., Chang, C. C., & Lin, C. J.* (2003). A practical guide to support vector classification. National Taiwan University, Taipei, Taiwan.
[8] *Nguyen, T., Park, EA., Han, J., Park, DC., Min, SY*. (2014). Object Detection Using Scale Invariant Feature Transform. In: Pan, JS., Krömer, P., Snášel, V. (eds) Genetic and Evolutionary Computing. Advances in Intelligent Systems and Computing, vol 238. Springer, Cham. https://doi.org/10.1007/978-3-319-01796-9_7
[9] *Calonder, M., Lepetit, V., Strecha, C., & Fua, P.* (2010). BRIEF: Binary robust independent elementary features. In European Conference on Computer Vision (pp. 778-792). Springer.
[10] *Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M.* (2020). YOLOv4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.
[11] *Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M.* (2021). YOLOv5: Improved real-time object detection. arXiv preprint arXiv:2104.02123.

ПОРІВНЯННЯ МЕТОДУ ОПОРНИХ ВЕКТОРІВ У ПОЄДНАННІ З АЛГОРИТМАМИ ВИЯВЛЕННЯ ОСОБЛИВИХ ТОЧОК ТА НЕЙРОМЕРЕЖ ДЛЯ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ

В.-Т. Лучка, В. Вдовиченко, Ю. Фургала

Львівський національний університет імені Івана Франка,
вул. Драгоманова, 50, Львів, 79005, Україна
VasylTaras.luchka@lnu.edu.ua

Виявлення об'єктів є одним із важливих завдань у галузі комп'ютерного зору, а прогрес у розвитку як традиційних алгоритмів виявлення особливих точок, так і нейромереж значно сприяє його вдосконаленню. У роботі порівняно та оцінено ефективність використання методу опорних векторів (SVC) разом із традиційними алгоритмами виявлення особливих точок, зокрема SIFT, ORB, BRISK, FREAK, із найновішим підходом на основі нейромереж, зокрема YOLOv5. За допомогою комплексного аналізу з'ясовано переваги та недоліки цих методів, виокремлено їхній потенціал для точного та ефективного

виявлення об'єктів. Для порівняння ефективності двох підходів до розпізнавання об’єктів на зображеннях проведено серію експериментів із визначеним набором зображень. Під час експерименту вибрано декілька параметрів для спостереження. Спочатку оцінено швидкодію алгоритмів SIFT, ORB, BRISK, FREAK на наборі даних, проаналізовано та порівняно кількість особливих точок, знайдених на зображені. Створено власну вирішальну функцію для розпізнавання об’єктів на зображенні, вхідними даними для якої були координати обраних особливих точок. Також створено модель класифікатора, що базується на методі опорних векторів із використанням особливих точок. Для порівняння результатів класифікатору на методі опорних векторів натреновано модель нейронної мережі на обраному наборі зображень. Для ефективної репрезентації можливостей нейронної мережі використано 3 моделі з різним розміром та різною кількістю епох тренування. Зважаючи на те, що метод опорних векторів (SVM) належить до парадигми контрольованого машинного навчання, встановлено різницю в ефективність і швидкість роботи із моделями нейромережі YOLOv5, а також час, потрібний для навчання моделей.
Ключові слова: ключові точки, метод опорних векторів, нейронні мережі, SIFT, ORB, BRISK, FREAK, YOLOv5

Стаття надійшла до редакції 05.03.2024
Прийнята до друку 23.03.2024