ISSN 2224-087X. Електроніка та інформаційні технології. 2022. Випуск 20. С. 51–61
Electronics and information technologies. 2022. Issue 20. P. 51–61

УДК 004.93 DOI: https://doi.org/10.30970/eli.20.6

GET A LIST OF FEATURE EXTRACTORS BASED ON FEATURE IMPORTANCE TECHNIQUES

M. Lyashkevych, V. Lyashkevych, R. Shuvar

System Design Department,
Ivan Franko National University of Lviv,
50 Drahomanova St., 79005 Lviv, Ukraine
Vasyl.Liashkevych@lnu.edu.ua

It is well-known, we live in the digital industrial era and we have a lot of accumulated volumes of structured and unstructured data. The accumulated data, actually, can help us make our solutions more efficient in different areas of human activities. That is why we are trying to recognize which kind of data can resolve which kind of problem. For this purpose, we have a lot of useful techniques for assessment of the data feature importance.

We are more confident in feature importance techniques when we work with numerical values of structured data because the detected objects are described by particular values. Alas, when extracting the features from unstructured data, like from images, it is still a question of how feature importance techniques are useful because it is either a question of how we are going to describe a detected object or which set of features might be an optimal one.

Of course, it is a good case when we can extract a lot of different features of a detected object but how it is enough if we want to recognize it from a list of other possible objects? This problem we are going to resolve this by an approach where we describe the nature of the detected object via its unique features either than feature importance techniques.

In the article, we consider the state of art feature importance methods, feature extraction algorithms and methodology of how to create the optimal set of feature extractors.

Keywords: Computer vision, object detection, feature importance, machine learning, water pollution, liquid contamination.

Introduction. Object detection is one of the well-known and widespread challenges in the industry which is being resolved by the latest achievements in computer vision (CV) with machine learning (ML) or deep learning (DL) techniques. Of course, the latest DL approaches are more powerful than classical ML algorithms because they use automatically trained feature detectors instead of ML algorithms. For ML algorithms we still should prepare the feature vectors. The feature extractors are prepared manually and it is the main reason why DL solutions are more accurate and stable in object detection problems. The water pollution detection by CV is the same problem where DL approaches are pretty good to apply but they have extra requirements for physical devices to run. Sometimes we can over these requirements but sometimes we cannot for different reasons like, for example, processor and memory limitations, energy safeness, etc.

Assuming we are working with small Edge AI devices which have an autonomous battery and could be allocated independently in different locations, for example, on the bottom of the

water tank, in dangerous environments with electromagnetic fields or radioactive zones. In this case, the heavy DL solutions are not effective because of energy consumption limitations.

Water or any other liquid pollution detection is so complex problem because it is so difficult to describe and annotate the nature of the problem and measure the level of the contamination using manual-created feature extractors. In the paper, we are going to show our approach and the results of executed experiments.

Observation of the latest approaches for solving the water pollution detection problem. The water pollution problem impacts our lifestyle and health. The boundaries of the problem are huge and generally could be classified into different types [1] where we would highlight the pollution of industrial wastewater for our investigation.

Actually, there are a lot of different types of water pollution and each type has its own reasons to be. Among the top 10 widespread types of pollution [2], we can notice that chemical pollution is a major part of them. The surface water and point source types of pollution are the most dangerous and could be detected by CV [3] with ML algorithms.

At the moment there are some efficient Waste Water Monitoring systems [4] which use different sensor-based approaches for pollution detection and monitoring. We can consider the system as an Early Warning system [5] because they alert us if a dangerous case is being detected. The systems can detect the chemical state in the water even if we can see visually the pollution but by using a visual channel for inspection, we can increment the quality of these kinds of systems. The visual channel shows the nature of the pollution which we can recognize visually but other sensors can recognize another nature of the same pollution and together they can recognize the pollution by different pollution natures. This approach could increase the quality of root cause analysis and generate more accurate signals in the Early Warning system.

Target types of pollution have no clear boundaries for annotation and we cannot crystal clearly detect small areas of the pollution by boundary boxes. Therefore, we are going to recognize the types and levels of pollution based on manually-prepared feature extractors. The main problem here is how to get a list of the most applicable feature extractors.

Water pollution definition. There are a lot of water pollution reasons [6] but only three broad categories of water pollution measurements: physical, chemical and biological. Unfortunately, none of them suits us because we are going to analyze by visual channel. Therefore, we defined three water pollution types empirically based on water pollution inspection:

- the category “turbidity”, contains very small particles which we cannot see with our eyes;
- the category “muddy”, contains special fluids which change water consistency and muddiness;
- the category “solid”, contains non-filterable divided solids which we can see with our eyes;

From the proposed approaches [7] for the water pollution level measurement, we can choose the parameter turbidity [8] which might be measured by a nephelometer or turbidimeter. These devices are good to use when we resolve a regression problem either than classification. Resolving the classification problem, we defined 5 levels of visually detected water pollution:

- level 1, clear water without any pollution. This level could be different for different water environments;
- level 2, is slight turbidity but all the objects in the background could be seen clearly;
- level 3, is turbidity where we cannot see the edges of the objects in the background clearly;

- level 4, is pollution where we see the not transparent areas but they are distributed in the image.
- level 5, is pollution where we cannot recognize the objects in the background.

The investigation of the existing data sets showed that we have no data set with criteria defined for our Early Warning system but we can use some samples from others with some corrections in annotations. We prepared a list of samples in the range from level 1 to level 5 with all types of pollution. The idea is to have a range of samples for visual comparison of different feature extractors. It is a useful approach in the initial stages of discovery. Below in fig. 1, we can see the samples which were used for the experiments.

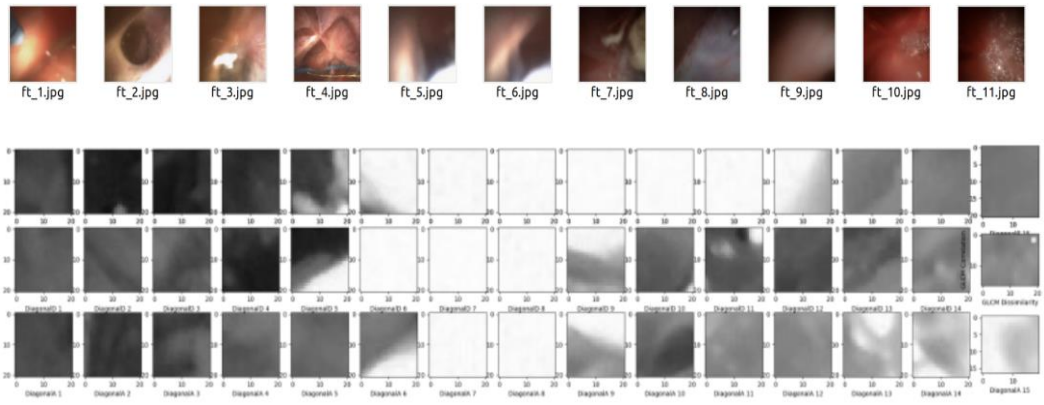


Fig. 1. The levels of water pollution

In the samples, we have different cases of water pollution for visual inspection of the efficiency of the feature extractors. These experiments give us intuition on how to group or asses feature extractors.

Measurement criteria. Analyzing the outputs of feature extractors we can do some remarks about the nature of the pollution:

- Blur - is the main part of turbidity nature. All other different kinds of extractors compensate for the particle's turbidity nature.
- Kernel-based / convolutional-based features are perspective options which could be very helpful but still, we need to investigate this option. Possibly, it could be the solution for extra blurring description and detection.
- Colour-based features show a good impact but in HCV format where each channel could be considered independently.
- Statistics-based features influence the results of prediction and help to split fluid turbidity from particle one.
- We have seen fewer impacts of contour-based features but still useful.

The results of the experiments are shown in fig. 2.

Basically, we prepared over 100 different feature extractors which can detect over 600 features from the images. Some of them could be discovered visually like different filters or other kinds of image manipulations (fig. 3). Applying different filters, we have different results which should be processed.

	c_BackProjection		c_EntropyShannon		c_GammaCorrection		c_UniformFiltering	
	value	rank	value	rank	value	rank	value	rank
samples/ft_1.jpg	9.312775	21	7.056964732	13	97.803675	9	99.1110375	17
samples/ft_2.jpg	11.90471875	18	7.06382241	12	103.1914729	5	97.613325	31
samples/ft_3.jpg	1.1301375	40	7.253616404	2	100.4186083	6	96.2997875	37
samples/ft_4.jpg	1.02559375	44	6.741923246	18	109.8334667	1	97.410375	34
samples/ft_5.jpg	14.3691125	15	7.160760019	6	105.6131042	3	93.66601458	49
samples/ft_6.jpg	16.63958125	13	7.101865818	10	109.2029917	2	94.14169792	46
samples/ft_7.jpg	7.097981198	25	6.506074629	23	66.41096032	32	100.7028083	7
samples/ft_8.jpg	79.52924609	1	6.643398523	20	83.54529032	18	99.66619617	12
samples/ft_9.jpg	1.101153786	41	6.71051911	19	75.11003488	28	98.77033047	23
samples/ft_10.jpg	1	47	6.351422924	28	71.78408914	29	100.9994777	6

Fig. 2. Results of experiments

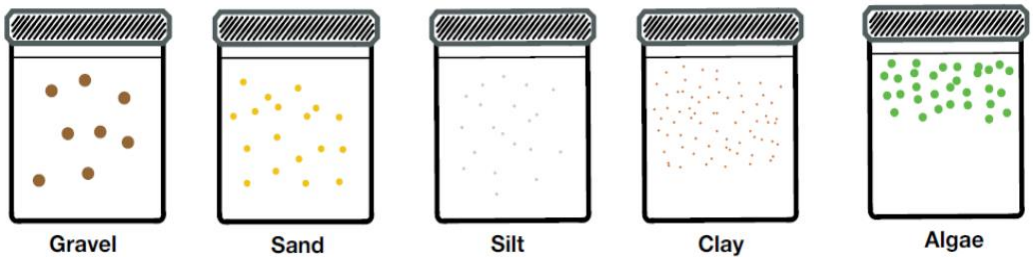


Fig. 3. Types of physical turbidity

Also, we prepared a lot of Gabor filters as it is shown in fig. 4. We can use one Gabor filter or use a collection of them. It depends on the requirements.

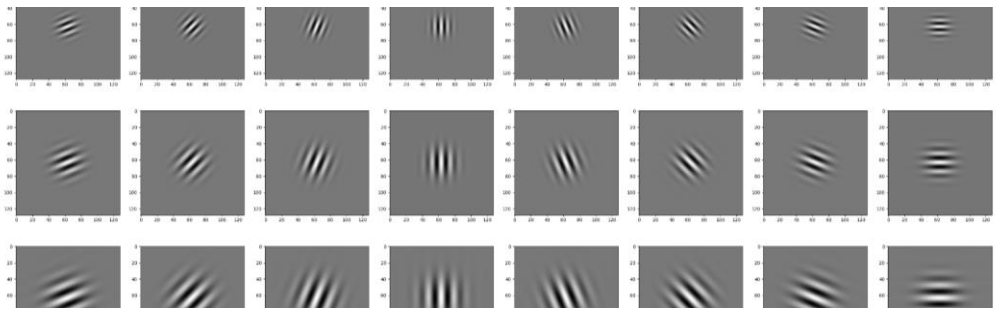


Fig. 4. Collection of Gabor filters

The more filters we have the more the training process of the ML model becomes difficult for different reasons. The big problem here is how to balance the feature extractors and how to

choose the optimal set of feature extractors. The scikit-learn framework has appropriate tools for feature importance investigation. For example, we can see the results of the investigation of some feature extractors in fig. 5.

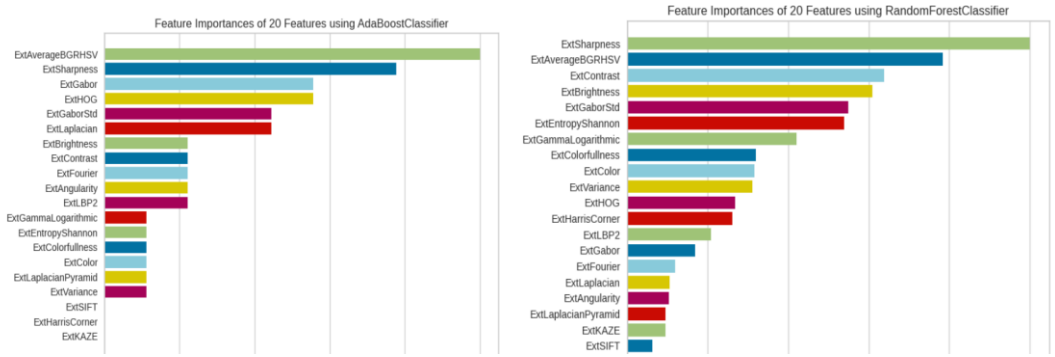


Fig. 5. Feature importance with scikit-learn

Pair analysis is one of the powerful techniques how to choose the best feature extractors. We use Pair-plot from the seaborn python-based library. The results can be seen in fig. 6-7.

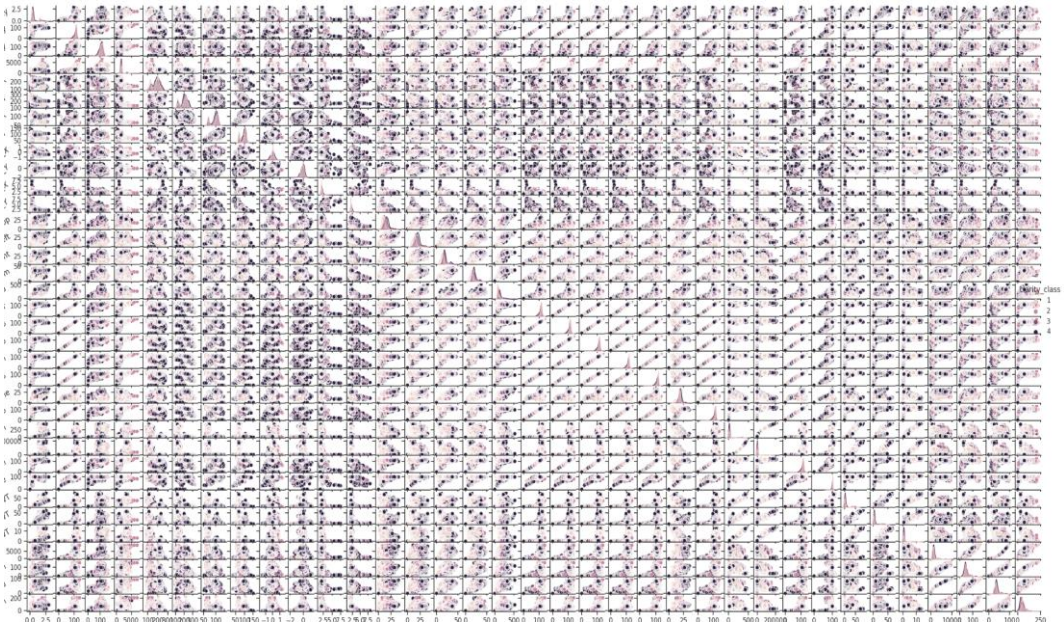


Fig. 6. Pair-plot for 32 feature extractors pair analysis

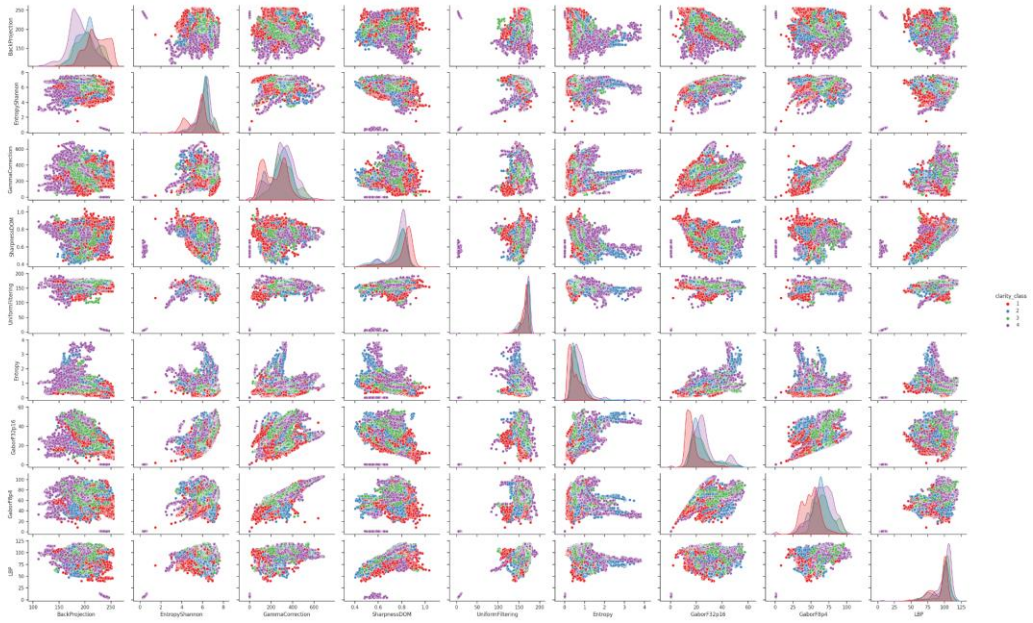


Fig. 6. Pair-plot for 9 feature extractors pair analysis

Motion Blur is present in several frames. As a model works with independent frames, the predicted level of turbidity is high due to a Motion Blur.

```
self.aOps.RandomBrightness(probability=1.0, min_factor=0.5, max_factor=1.5),
self.aOps.RandomColor(probability=1.0, min_factor=0.5, max_factor=1.5),
self.aOps.RandomContrast(probability=1.0, min_factor=0.5, max_factor=1.5),
self.aOps.Skew(probability=1.0, skew_type='TILT', magnitude=8), # -> to exclude
self.aOps.Rotate(probability=1.0, rotation=90),  # one of the best
self.aOps.Rotate(probability=1.0, rotation=270),
self.aOps.RotateRange(probability=1.0, max_left_rotation=25, max_right_rotation=25),
self.aOps.Zoom(probability=1.0, min_factor=1.1, max_factor=1.6),
self.aOps.Flip(probability=1.0, top_bottom_left_right='LEFT_RIGHT'),   # one of the best
self.aOps.Flip(probability=1.0, top_bottom_left_right='TOP_BOTTOM'),
self.aOps.CropRandom(probability=1.0, percentage_area=0.5),  # -> to exclude
self.aOps.Resize(probability=1.0, width=250, height=250, resample_filter='NEAREST'),   # one of the best
self.aOps.Shear(probability=1.0, max_shear_left=20, max_shear_right=20),
self.aOps.Scale(probability=1.0, scale_factor=1.5),
self.aOps.Distort(probability=1.0, grid_width=8, grid_height=8, magnitude=8),
self.aOps.GaussianDistortion(probability=1.0, grid_width=8, grid_height=8, magnitude=8,
                             corner="bell", method="in", mex=1.0, mey=1.0, sdx=1.0, sdy=1.0),
self.aOps.Zoom(probability=1.0, min_factor=1.1, max_factor=1.6),
# self.aOps.ZoomRandom(probability=1.0, percentage_area=0.75, randomise=True),
# self.aOps.HSVShifting(probability=1.0, hue_shift=20, saturation_scale=20, saturation_shift=8,
#                       value_scale=15, value_shift=8),
self.aOps.RandomErasing(probability=1.0, rectangle_area=0.5)
```

Fig. 7. Data augmentation algorithms

Augmentation approaches were investigated too. Initially, we chose 18 algorithms for augmentation but finally applied only 3 of them, fig. 7. The best options for augmentation are rotation by 90, left-right flip and resizing option (250x250).

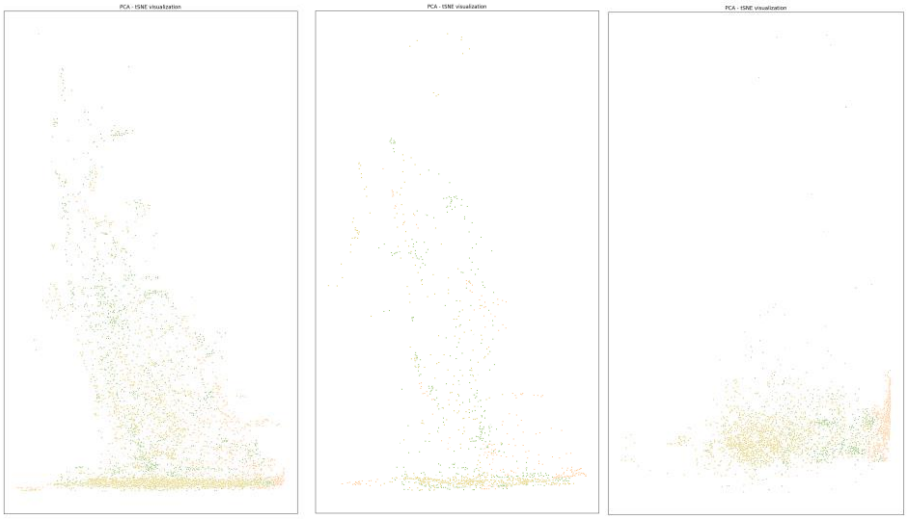


Fig. 8. Data set visualization: train set on your left, validation set on the centre, and test set on your right

The correlation between features we did in two ways: using a correlation matrix (fig. 9) and using hierarchical clustering (fig. 10).

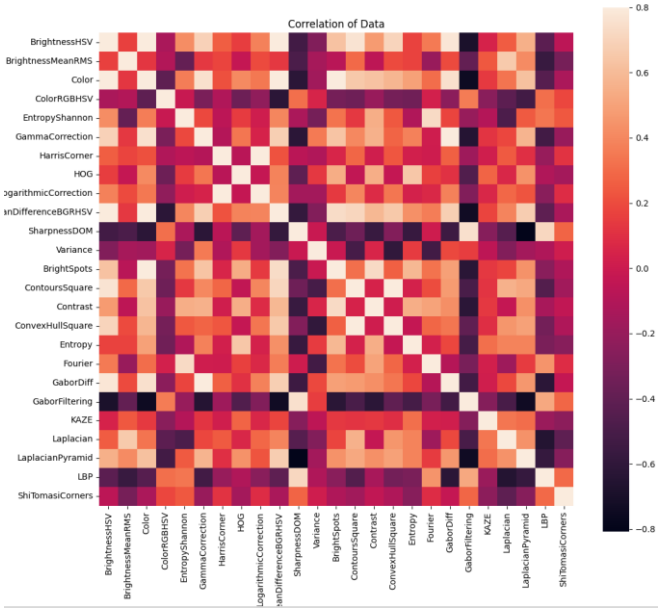


Fig. 9. Correlation matrix

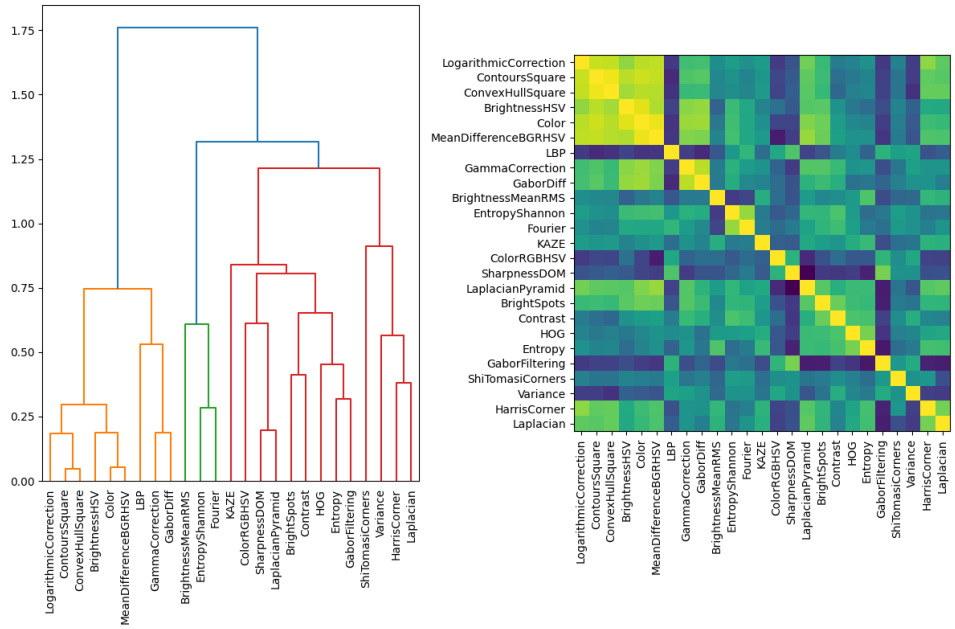
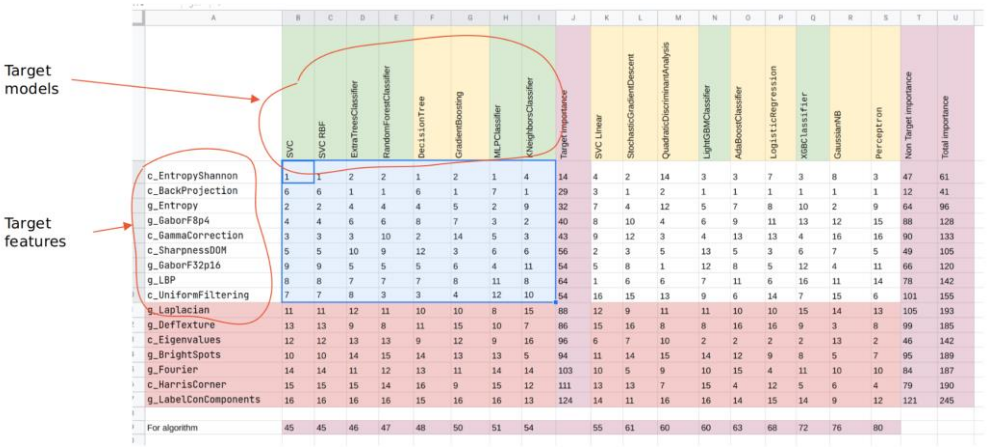


Fig. 10. Hierarchical clustering

Feature-importance experiments were planned using a list of prepared samples. We choose over 15 ML algorithms trained them with all feature extractors and did the feature importance. As result, we assess and range the results for the next iteration of the experiments. In some iterations, we came up with the optimal list of algorithms and a list of the best-fit features for them (fig. 11).



	SVC	SVC RBF	ExtraTreesClassifier	RandomForestClassifier	DecisionTree	GradientBoosting	MLPClassifier	KNeighborsClassifier	Target importance	SVC Linear	StochasticGradientDescent	QuadraticDiscriminantAnalysis	LightGBMClassifier	AdaBoostClassifier	LogisticRegression	XGBClassifier	GaussianNB	Perceptron	Non Target importance	Total importance
c_EntropyShannon	1	1	2	2	1	2	1	4	14	4	2	14	3	3	7	3	8	3	47	61
c_BackProjection	6	6	1	1	6	1	7	1	29	3	1	2	1	1	1	1	1	1	12	41
g_Entropy	2	2	4	4	4	5	2	9	32	7	4	12	5	7	8	10	2	9	64	96
g_GaborF8p4	4	4	6	6	8	7	3	2	40	8	10	4	6	9	11	13	12	15	88	128
c_GammaCorrection	3	3	3	10	2	14	5	3	43	9	12	3	4	13	13	4	16	16	90	133
c_SharpnessDOM	5	5	10	9	12	3	6	6	56	2	3	5	13	5	3	6	7	5	49	105
g_GaborF32p16	9	9	5	5	5	6	4	11	54	5	8	1	12	8	5	12	4	11	66	120
g_LBP	8	8	7	7	7	8	11	8	64	1	6	6	7	11	6	16	11	14	78	142
c_UniformFiltering	7	7	8	3	3	4	12	10	54	16	15	13	9	6	14	7	15	6	101	155
g_Laplacian	11	11	12	11	10	10	8	15	88	12	9	11	11	10	10	15	14	13	105	193
g_DefTexture	13	13	9	8	11	15	10	7	86	15	16	8	8	16	16	9	3	8	99	185
c_Eigenvalues	12	12	13	13	9	12	9	16	96	6	7	10	2	2	2	2	13	2	46	142
g_BrightSpots	10	10	14	15	14	13	13	5	94	11	14	15	14	12	9	8	5	7	95	189
g_Fourier	14	14	11	12	13	11	14	14	103	10	5	9	10	15	4	11	10	10	84	187
c_HarrisCorner	15	15	15	14	16	9	15	12	111	13	13	7	15	4	12	5	6	4	79	190
g_LabelConComponents	16	16	16	16	15	16	16	13	124	14	11	16	16	14	15	14	9	12	121	245
For algorithm	45	45	46	47	48	50	51	54		55	61	60	60	63	68	72	76	80		

Fig. 11. The results of iterative feature extractors investigation

The results of feature importance with scikit-learn for XGBoost Classifier (fig. 12).

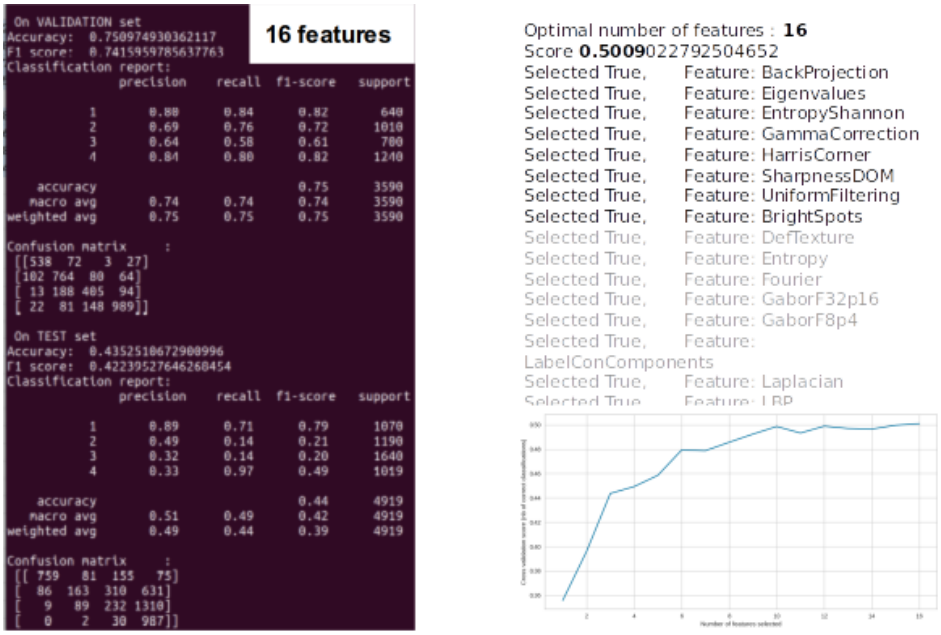


Fig. 12. XGBoost Classifier

All results of the testing are shown in fig. 13. You can see apart for validation and for the test set. Here is shown result for 16 features and 17 ML algorithms. The results of speed testing is shown in fig. 14.

	16 FEATURES			
	Validation set		Test set	
	Accuracy	F1 Score	Accuracy	F1 Score
ExtraTreesClassifier	0.7846796657	0.7717460239	0.4378938809	0.4032421263
KNeighborsClassifier	0.7515320334	0.7471187695	0.46452531	0.4658050473
SVC RBF	0.752367688	0.7398416415	0.4769262045	0.4668004926
LightGBMClassifier	0.7259052925	0.7167150423	0.4364708274	0.4297299316
MLPClassifier	0.7353760446	0.7282975392	0.4157349055	0.3899099027
RandomForestClassifier	0.7534818942	0.7418505556	0.4315917869	0.4159638738
SVC	0.752367688	0.7398416415	0.4769262045	0.4668004926
XGBClassifier	0.7509749304	0.7415959786	0.4352510673	0.4223952765
GradientBoosting	0.6707520891	0.6507026197	0.4419597479	0.4297710603
DecisionTree	0.6200557103	0.6094482468	0.4082130514	0.4146279248
SVC Linear	0.5894150418	0.5759200013	0.4879040455	0.4826170201
StochasticGradientDescent	0.5473537604	0.5172458554	0.4326082537	0.4170913617
LogisticRegression	0.5442896936	0.5226868863	0.4592396829	0.4450116637
QuadraticDiscriminantAnalysis	0.5323119777	0.5512736445	0.3262858305	0.3322134888
AdaBoostClassifier	0.5292479109	0.5138256234	0.4718438707	0.472097973
Perceptron	0.443454039	0.464814328	0.3073795487	0.3132070972
GaussianNB	0.3635097493	0.3678343891	0.3771091685	0.3713857109

Fig. 13. The results of the training

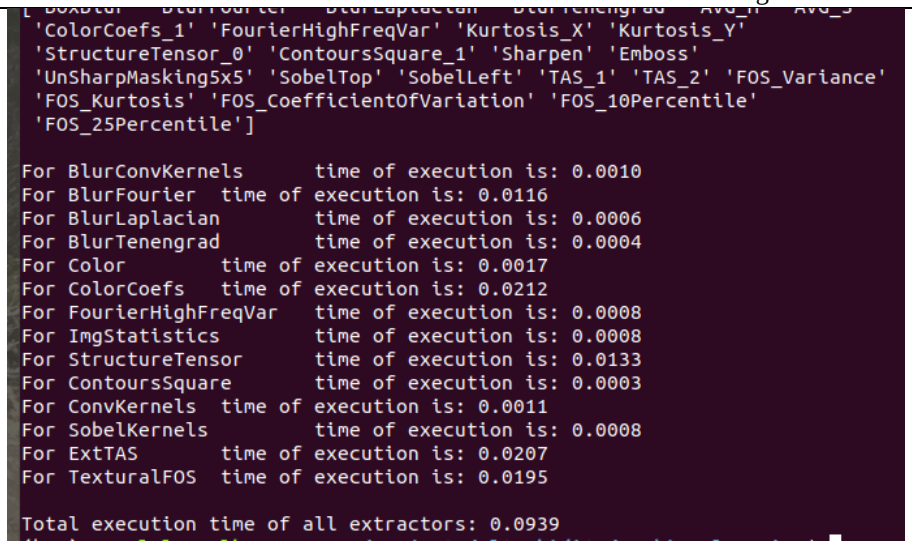


Fig. 14. Testing results of feature extractors speed

When we have the speed of the extractors, we can pick up the best fit set of features based on speed requirements.

Summary. Water quality monitoring is a crucial aspect of protecting water resources. Under the Clean Water Act, state, tribal and federal agencies monitor lakes, streams, rivers and other types of water bodies to determine water quality conditions. The data generated from these monitoring activities help water resource managers know where pollution problems exist, where to focus pollution control energies [9] and where progress has been made.

Our solution is called to enhance the existing solutions built based on chemical sensors whereby extra visual channels are driven by CV. Also, it could be possible to do this in dangerous electromagnetic waves and radiational areas.

References

[1] *Melissa Denchak*. Water Pollution: Everything You Need to Know. - 2022. - Mode access: https://www.nrdc.org/stories/water-pollution-everything-you-need-know

[2] Linquip Team.10 Types of Water Pollution in 2022 + PDF. - 2021. - Mode access: https://www.linquip.com/blog/types-of-water-pollution/

[3] *Lærke Isabella, Nørregaard Hansen*. Turbidity measurement based on computer vision. - Mode access: https://projekter.aau.dk/projekter/files/306657262/master.pdf.

[4] S::can GmbH. Waste Water Monitoring. - 2021. - Mode access: https://www.s-can.at/en/applications/waste-water/?gclid=CjwKCAiAp7GcBhA0EiwA9U0mtgPirdjVNR2UTTnYmKscQzG7uUbTYCJX-ToT2ADVESgftwG2tebxURoCOyEQAvD_BwE

[5] *Ding, X.; Zhang, J.; Jiang, G.; Zhang, S.* Early Warning and Forecasting System of Water Quality Safety for Drinking Water Source Areas in Three Gorges Reservoir Area, China. Water 2017, 9, 465. https://doi.org/10.3390/w9070465

[6] Wikipedia. Water pollution. - Mode access: https://en.wikipedia.org/wiki/Water_pollution

[7] *Umair Ahmed, Rafia Mumtaz, Hirra Anwar, Sadaf Mumtaz, Ali Mustafa Qamar*; Water quality monitoring: from conventional to emerging technologies. Water Supply 1 February 2020; 20 (1): 28–45. DOI: https://doi.org/10.2166/ws.2019.144

[8] *Patel J. Y. Vaghani M. V.* 2015 Correlation study for assessment of water quality and its parameters of par river Valsad, Gujarat, India. IJIERE 2, 150–156.

[9] Recent advances and challenges for water evaporation-induced electricity toward applications / Van-Duong Dao, Ngoc HungVu, Hai-LinhThi Dang, SiningYun // Nano Energy. - V. 85. - 2021. - Mode access: https://doi.org/10.1016/j.nanoen.2021.105979.

ОТРИМАННЯ СПИСКУ ВИДОБУВАЧІВ ОЗНАК НА ОСНОВІ МЕТОДІВ ОЦІНКИ ВАЖЛИВОСТІ ОЗНАК

М. Ляшкевич, В. Ляшкевич, Р. Шувар

Кафедра системного проектування,
Львівський національний університет імені Івана Франка,
вул. Драгоманова, 50, Львів, 79005, Україна
Vasyl.Liashkevych@lnu.edu.ua

Як відомо, ми живемо у цифровій індустріальній епосі і ми маємо багато накопичених структурованих і неструктурованих даних. Насправді, структуровані дані можуть допомогти нам зробити наші рішення більш ефективними у різних областях діяльності людини. Це тому ми намагаємося розпізнати які типи даних, які саме типи проблем розв'язують. З цією метою, ми маємо багато корисних методів для оцінки важливості ознак даних. Scikit-learn це є найбільш популярна бібліотека серед науковців, реалізована на мові Python, яка дозволяє досліджувати важливість ознак даних.

Зазвичай, науковці більш упевнені у методах важливості ознак коли вони працюють з числовими значеннями структурованих даних тому, що об'єкти, які розпізнаються, описані конкретними значеннями у таблицях. Нажаль, коли ми видобуваємо ознаки з неструктурованих даних, як наприклад рисунків, це ще є питання наскільки методи оцінки важливості ознак є корисними тому, що це є скоріше питання опису об'єктів, що розпізнаються чи яка множина ознак могла би бути оптимальною.

Насправді не можливо однозначно відповісти на питання скільки різних ознак з об'єктів, що детектуються, нам потрібно аби ми могли їх класифікувати з високою імовірністю. Наскільки це "достатньо" у випадку, коли ми хочемо розпізнати ці об'єкти зі списку інших об'єктів? Дану проблему ми збираємося вирішити підходом, коли ми описуємо фізичну природу об'єктів, що детектуються через їх унікальні ознаки скоріше ніж із використання методів оцінки важливості ознак. Методи оцінки важливості ознак є корисними для нас на початкових етапах, а також для оцінки отриманих результатів.

У статті, ми розглядаємо сучасні методи оцінки важливості ознак, алгоритми видобування ознак і методологію як саме створити оптимальну множину видобувачів ознак. Для того, щоб краще засвоїти наведені методики, ми розпізнавали у своїх дослідах рівень забруднення води, а також тип забруднення.

Ключові слова: Комп'ютерний зір, розпізнавання об'єктів, важливість ознак, машинне навчання, забруднення води, забруднення речовини.

Стаття надійшла до редакції 29.11.2022.
Прийнята до друку 05.12.2022.