Methodology of Machine Learning Model Development for Solving Applied Computer Vision Problems

Marina Lukashevich
Information Management Systems Department
Belarusian State University
Minsk, Belarus
LukashevichMM@bsu.by

***Abstract*—The methodology of machine learning model development for solving applied computer vision problems is presented. The article discusses the tasks of computer vision, the main components of building application systems and the challenges and limitations of the existing technological level.**

***Keywords*—Machine learning, computer vision, machine learning model, image, video**

I. Introduction

Machine learning (ML) is the creation and application of models internalized from data. In the case of traditional programming, rules are expressed in a programming language. They act on data, and computer programs provide answers. In the case of machine learning, the answers (typically called labels) are provided along with the data, and the machine infers the rules that determine the relationship between the answers and the data. Machine learning involves algorithms that learn from patterns of data and then apply them to decision making (Figure 1).

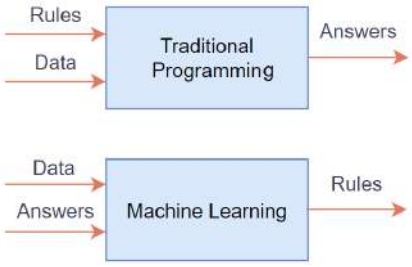

Figure 1. Machine learning vs. traditional programming

Machine learning can also be defined as the process of solving a practical problem by:

- collecting a dataset;
- algorithmically training a statistical model on that dataset.

Machine learning does not have a clear sequence of steps because it is necessary to work with different types of data (tabular data, images, video, signals, text, speech, etc.) and in different domains (data analysis, computer vision, natural language processing, robotics, etc.). Each case has its own specifics, algorithmic techniques, and tools. The main goal of this paper is to summarize theoretical background and practical experience, formalize a methodology for machine learning model building for solving computer vision problems, and formulate some practical recommendations.

II. Computer vision analysis

A. Computer vision

Computer vision (CV) is defined as the automatic extraction of information from images or videos. Some tasks require computer vision to simulate human vision. In other cases, it is necessary to perform statistical data processing, geometric transformations, etc. In practice, computer vision is a fusion of artificial intelligence, pattern recognition, digital signal and image processing, math, and physics (Figure 2). It depends on the specific problem being solved.

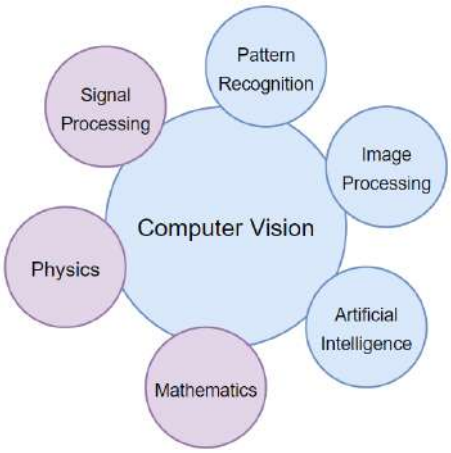

Figure 2. Interdisciplinarity of computer vision

B. Object and scene level tasks

The focus of object-level tasks is on objects in a visual scenario, and they require the analysis and understanding of various entities or instances that are associated with them. These tasks involve recognizing objects, detecting them, tracking them, detecting changes, detecting anomalies, and segmenting them.

The process of object recognition involves identifying and categorizing objects into predetermined classes or categories. In object detection, the procedure of recognizing and precisely locating an object within an image or video is performed by creating a bounding box around it.

The process of tracking the movement of objects across multiple frames of aerial video or image sequences is referred to as object tracking. The process of identifying alterations in imagery over different time instances is called change detection.

The process of systematic identification of abnormal patterns or objects within visual data and contrasting them with established norms is referred to as anomaly detection.

In semantic segmentation, semantic labels or classes are assigned to each pixel in an image. In the case of instance segmentation, semantic labels, or class labels, are assigned to each pixel in an image with distinction between individual instances of the same class. The complexity and list of computer vision tasks for object level are presented in Figure 3.

Scene-level tasks focus on a specific scene or an entire image or video scenario in visual data and involve in-depth analysis of context, composition, or environmental features. Tasks such as image registration, 3D reconstruction and terrain modeling, and localization and mapping are some of the scene-level tasks. The complexity and list of computer vision tasks for scene level are presented in Figure 4.

C. Revolution in computer vision

In 2006, Nvidia released CUDA, a programming language that allowed GPUs to be used as general-purpose supercomputers. In 2009, artificial intelligence researchers at Stanford introduced Imagenet, a collection of labeled images used to train computer vision algorithms. A revolution in computer vision occurred when neural networks aimed at working with images began to be used. These are called convolutional neural networks. In 2012, convolutional neural networks (AlexNet) significantly reduced the error in classification and approached the result, which shows in image recognition by a human about 5% of errors. And in 2015, neural networks overtook humans in recognition accuracy and showed a result of 3.6%. CNN, the ImageNet dataset, and graphics processors were the magic combination that launched a powerful advance in computer vision. In the last few years, neural networks based on transformer architecture with Attention mechanism have shown excellent efficiency in computer vision.

The most effective solutions in the field of computer vision are based on neural networks in general and on deep neural networks in particular. A large number of effective architectures of deep neural networks have been proposed by researchers. These architectures are implemented in modern deep learning frameworks and are widely used in practice. The classical theory of pattern recognition and digital image processing began to take a back seat. The classical scheme, including image preprocessing, feature computation, and decision-making, has become less effective. The use of neural networks involves feature computation and decision making by the neural network itself [1]–[8].

III. Components for solving computer vision problems

Highlight the main components necessary for realizing practical solutions for machine learning tasks.

- Datasets.
- Frameworks and libraries.
- Model architecture.
- Hardware resources.

Machine learning models are built on the basis of data. Two types of datasets can be identified:

- large datasets used for model pre-training and implementation of transfer learning technology (for example, ImageNet, COCO, etc.);
- custom datasets are collected and labeled for a specific task.

It is a good solution for scientific and educational tasks to use public datasets from well-known platforms (Kaggle, Roboflow, etc.). In addition, it should be mentioned that synthetic data can be used to train models in some cases. Synthetic data can be obtained in the following ways:

- using generative neural networks;
- building 3D models of objects, creating synthetic images and videos with different backgrounds, extraneous objects, etc. on their basis (using Blender, NVidia Omniverse, etc.).

There are two leading frameworks for deep neural network development: PyTorch and TensorFlow. Both are powerful frameworks with unique strengths. PyTorch is favored for research and dynamic projects, while TensorFlow excels in large-scale and production environments. Industry experts may recommend TensorFlow, while hardcore ML engineers may prefer PyTorch. However, there has been a general trend of increasing usage and preference for Pytorch.

In addition to frameworks, a number of libraries and IDEs are used in building computer vision solutions, as well as in experiments and development processes. Some of them are presented in Table I.

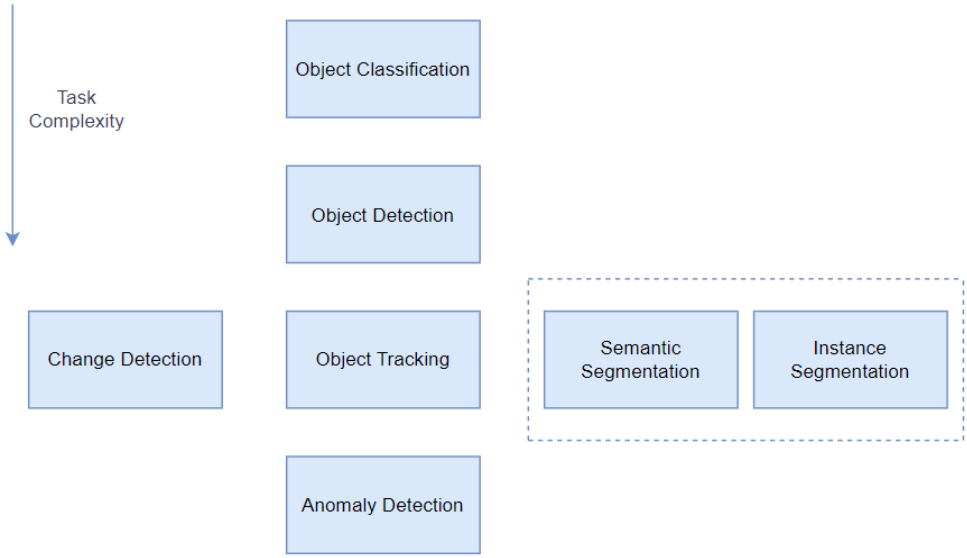

Figure 3. Complexity of computer vision tasks. Object level tasks

Table I
Some ML/CV frameworks and libraries

Data	**Training/Evaluation**	**Libraries for ML/CV**
Labeling CVTA, LabelMe, COCO-Annotator, COCO-ui	**Frameworks and Distributed Training** TensorFlow, Keras, TensorFlow Lite, PyTorch, PyTorch Lightning	**Machine Learning** Numpy, Scipy, Scikit-learn, Pandas, Matplotlib, Seaborn, h5py
Exploration Pandas, Seaborn	**Software Engineering** Python, PyCharm, VS Code, git	**Computer Vision, Digital Image Processing** Scikit-image, OpenCV, Pillow, Matplotlib
	Experiment Management TensorBoard, Weights and Biases	
	Hyperparameter Tuning Weights and Biases	

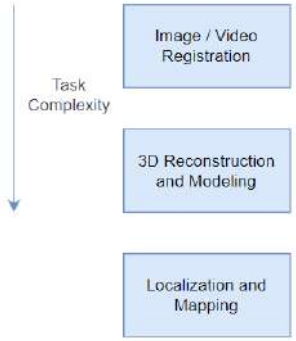

Figure 4. Complexity of computer vision tasks. Scene level tasks

When selecting a neural network architecture to solve a given problem, there are several options. As a rule, while working on projects, researchers and developers apply to each of them in the sequence in which they will be listed:

- state-of-the-art (SOTA) architecture for specific computer vision problems (classification, detection, segmentation, etc.);
- modified state-of-the-art architecture for specific computer vision problems (classification, detection, segmentation, etc.);
- costume state-of-the-art architecture for specific computer vision problems (classification, detection, segmentation, etc.).

Some state-of-the-art architectures for computer vision tasks are presented in Table II. This is by no means a complete list, but it will provide insight into the variety of neural network architectures for solving computer vision problems.

The architecture selection process should also be guided by the model size and its inference time. There may be situations when it is necessary to develop a lightweight model, for example, for cutting-edge devices, and to provide a high speed of inference.

Another necessary component for the development of computer vision solutions is computational resources. It is possible to use both GPUs on workstations and cloud computing resources (for example, Amazon Web Services, Google Cloud Platform, Microsoft Azure, etc.).

Table II
Some State-of-the-art NN architectures for computer vision

Task	SOTA Architectures
Classification	Xception, VGG16, VGG19, ResNet50, ResNet101, ResNet152, InceptionV3, MobileNet, MobileNetV2, DenseNet121, DenseNet169, DenseNet201, NASNetMobile, NASNetLarge, EfficientNetB0,EfficientNetB1, EfficientNetB2, EfficientNetB3, EfficientNetB4, EfficientNetB5, EfficientNetB6, EfficientNetB7, ConvNeXTTiny, ConvNeXTSmall, ConvNeXTBase, ConvNeXTLarge
Detection	SSD, YOLO5, YOLO7, YOLO8, YOLO9, YOLO NAS
Segmentation	Unet, FPN, Linknet, PSPNet, SegFormer

NVIDIA, the market leader, offers deep-learning GPUs.

IV. Data Importance

The main blockers in machine learning projects are data unavailability, an insignificant number of data, and low variability. Often, the choice of neural network architecture is not as critical as an underpowered dataset. Also, data collection and labeling for model building can take a significant amount of time. Especially recently, much attention has also been paid to personal data, the impossibility of using it, and data ethics in general.

Formulate some rules about data for ML model development:

- at a bare minimum, collect around 1000 examples.
- for most "average" problems, collect 10,000 - 100,000 examples.
- for “hard” problems like machine translation, high dimensional data generation, or anything requiring deep learning, collect 100,000 — 1,000,000 examples.

Generally, the more dimensions your data has, the more data you need. It is necessary to have roughly 10 times the amount of data in your examples. The more complex the problem, the more data you need.

A. Multiplicity of computer vision tasks and interdisciplinarity

We would like to emphasize that there is often a need for special, interdisciplinary knowledge when solving computer vision problems. This is evident when working with a class of medical images. In such cases, it is necessary to involve medical experts both for data labeling and for result interpretation. A good rule of thumb is to engage several experts at the same time and average their labels and estimates.

On the other hand, different computer vision problems are possible for the same class of images. It is very important to formulate the problem in terms of machine learning and computer vision (image classification, object detection, image segmentation, etc.) at the very beginning of the project. This will immediately give an understanding of how to label images (polygon, class label for a whole image, using bounding boxes, etc.) and with what architectures and algorithms to perform experiments (Table II).

As an example, here are images and possible computer vision tasks. This is a class of medical images. Figure 5 shows examples of retina images and their labels for classifying the stages of diabetic retinopathy. Figure 6 shows examples of optical disk detection. And Figure 7 shows the results of vessel segmentation in retinal images [9]–[11].

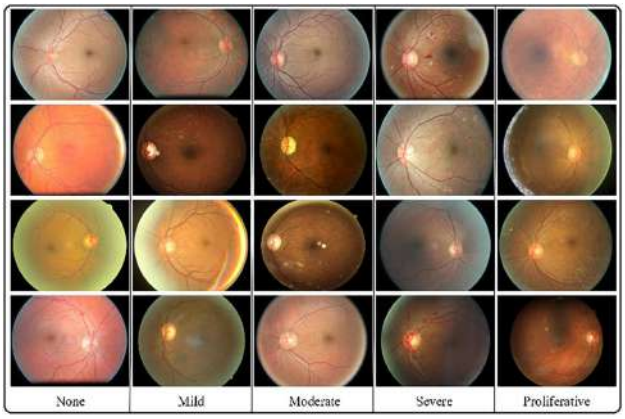

Figure 5. Diabetic retinopathy classification

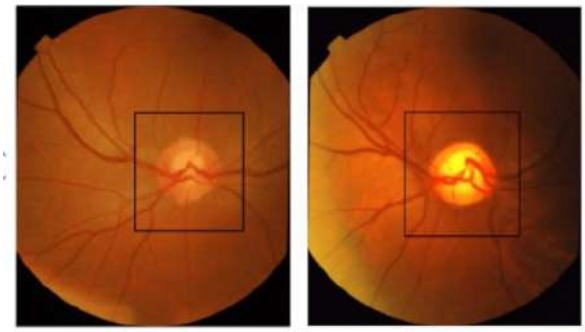

Figure 6. Optical disk detection

V. Methodology of machine learning model development for Computer Vision

Summarize the above information and formulate a methodology for machine learning model development for solving computer vision tasks. There are three logical levels to this methodology:

- coding.

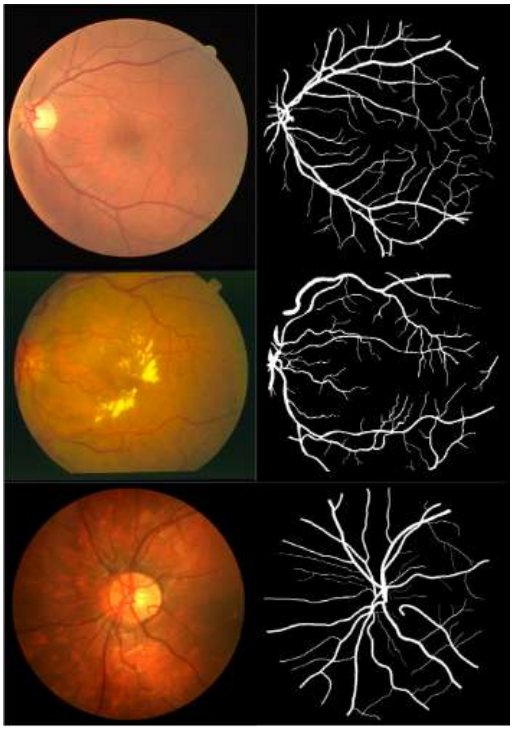

Figure 7. Retinal vessels segmentation

- model development.
- working with data.

The methodology includes next stages:

- *Model Building*. The computer vision task is formulated (classification, detection, segmentation, etc.), the stack of technologies used is determined, data is collected and labeled, promising model architectures are determined, and models are built. A common practice is to split the data set into a training set, a validation set, and a test set. The training dataset is used for model building. A validation dataset is necessary to improve the training process.
- *Model Evaluation and Experimentation.* The accuracy of the models is assessed on the test dataset based on metrics for specific tasks. The best model is chosen.
- *Productionize Model.* The model is saved using the selected format (ONNX, h5, etc.).
- *Testing*. Code and model are tested using training dataset.
- *Deployment.* The deployment of the model in the product environment is determined and implemented. There are several possible options:
  - batch deployment;
  - real time;
  - streaming deployment;
  - edge deployment.
- *Monitoring and Observability.* Continuously monitoring and testing the model's effectiveness post-deployment pose ongoing challenges. Regular monitoring is necessary to ensure accurate results, identify potential issues, and drive performance enhancements.

VI. Challenges of building intelligent systems for computer vision tasks

Despite the great theoretical and technical progress in the field of computer vision, there are a number of limitations that need to be overcome in the future. Let's name some of them.

- The diversity of visual representation, such as illumination, perspective, or occlusion in objects, is a major challenge. These variations must be overcome to eliminate any visual inconsistencies.
- With each image consisting of millions of pixels, dimensional complexity becomes another barrier to overcome. This could be done by using different techniques and methodologies.
- Real-time processing can be challenging. This comes into play when making decisions for autonomous navigation or interactive augmented realities, which require optimal performance of computational frameworks and algorithms for fast and accurate analysis.
- Ethical considerations are paramount in artificial intelligence, and computer vision is no different. This could be bias in deep learning models or any discriminatory results. This emphasizes the need for a proper approach to dataset curation or algorithm development.

Outline promising directions in the field of computer vision in the next few years: zero-shot learning, few-shot learning, and one-shot learning. It makes sense to develop scientific research in this direction. Zero-shot learning, few-shot learning, and one-shot learning are all techniques that allow a machine learning model to make predictions for new classes with limited labeled data. The choice of technique depends on the specific problem and the amount of labeled data available for new categories or labels (classes) [12].

If we move away from technical blockers and think about high-level analysis of computer vision challenges, then it is obvious that the component design of hybrid and intelligent systems will be promising in the future. It is necessary to solve the problem of the compatibility of scientific research results in the field of artificial intelligence. This problem is currently the key one preventing the active development of artificial intelligence.

a significant reduction in the effectiveness of using the component method. designing computer systems based on reusable libraries components.

Insufficiently high degree of learning about modern computer systems during their operation, which results

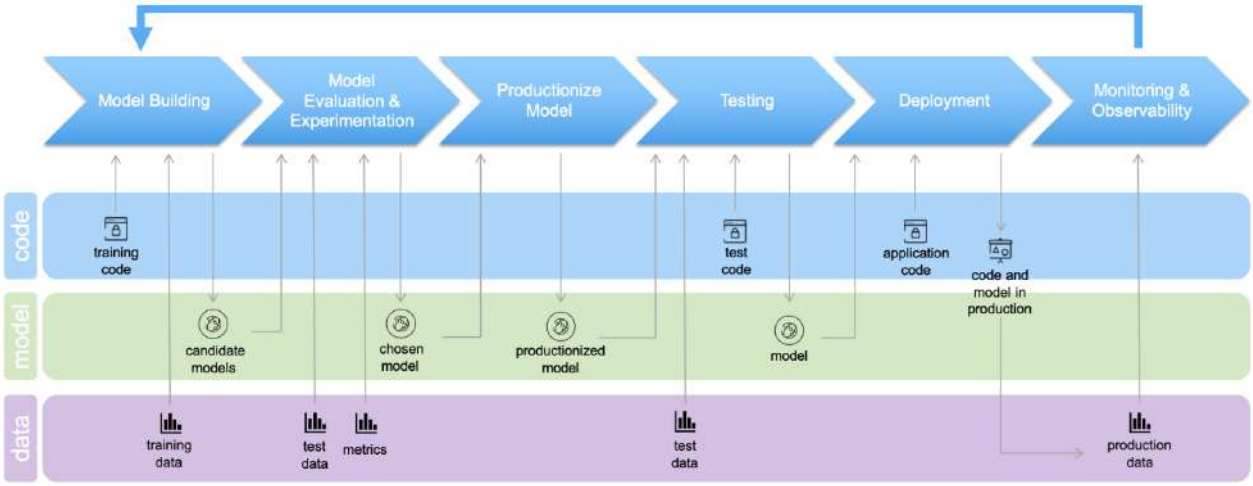

Figure 8. Methodology of machine learning model development for computer vision

in the high complexity of their maintenance and improvement, as well as their insufficiently long life cycle.

The problems of the unification of the principles for constructing various components of computer systems are solved in the OSTIS project. The OSTIS The project aims to create an open semantic technology for designing knowledge-driven systems in general and computer vision systems in particular [13].

VII. Conclusion

The methodology for developing a machine learning model for solving applied computer vision problems is presented. The article discusses the tasks of computer vision, the main components of building application systems, and the and the challenges and limitations of the existing technological level. demonstrated the need to develop areas related to the compatibility of scientific research results in the field of artificial intelligence.

References

[1] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. Communications of the ACM, 60(6):84–90, 2017.

[3] Sebastian Raschka; Yuxi (Hayden) Liu; Vahid Mirjalili; Dmytro Dzhulgakov, Machine Learning with PyTorch and Scikit-Learn: Develop machine learning and deep learning models with Python , Packt Publishing, 2022.

[4] Fermüller, Cornelia and Michael Maynord. “Advanced Methods and Deep Learning in Computer Vision.” (2022).

[5] Zhengxia Zou, Keyan Chen, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. Proceedings of the IEEE, 2023.

[6] Tianfei Zhou, Fatih Porikli, David J Crandall, Luc Van Gool, and Wenguan Wang. A survey on deep learning technique for video segmentation. TPAMI, 2023.

[7] Jingjing Xu, Wangchunshu Zhou, Zhiyi Fu, Hao Zhou, and Lei Li. A survey on green deep learning. arXiv preprint arXiv:2111.05193, 2021.

[8] Jianyuan Guo, Kai Han, Han Wu, Yehui Tang, Xinghao Chen, Yunhe Wang, and Chang Xu. Cmt: Convolutional neural networks meet vision transformers. In CVPR, pages 12175–12185, 2022.

[9] R. K. Kumar and K. Arunabhaskar, "A Hybrid Machine Learning Strategy Assisted Diabetic Retinopathy Detection based on Retinal Images," 2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES), 2021, pp. 1-6, doi: 10.1109/ICSES52305.2021.9633875.

[10] Dagliati, A, Marini, S, Sacchi, L, Cogni, G, Teliti, M, Tibollo, V, De Cata, P, Chiovato, L & Bellazzi, R 2018, ’Machine Learning Methods to Predict Diabetes Complications’, Journal of Diabetes Science and Technology, vol. 12, no. 2, pp. 295-302. https://doi.org/10.1177/1932296817706375

[11] Lukashevich M.M. A neural network classifier for detecting diabetic retinopathy from retinal images. «System analysis and applied information science». 2023;(1):25-34. (In Russ.) ttps://doi.org/10.21122/2309-4923-2023-1-25-34

[12] Labrak, Y., Rouvier, M., Dufour, R. (2023). A zero-shot and few-shot study of instruction-finetuned large language models applied to clinical and biomedical tasks. arXiv preprint arXiv:2307.12114.

[13] V. Golenkov, N. Guliakina, V. Golovko, V. Krasnoproshin, “Methodological problems of the current state of works in the field of artificial intelligence,” Otkrytye semanticheskie tekhnologii proektirovaniya intellektual’nykh system [Open semantic technologies for intelligent systems], pp. 17–24, 2021.

МЕТОДОЛОГИЯ РАЗРАБОТКИ МОДЕЛЕЙ МАШИННОГО ОБУЧЕНИЯ ДЛЯ РЕШЕНИЯ ПРИКЛАДНЫХ ЗАДАЧ КОМПЬЮТЕРНОГО ЗРЕНИЯ

Лукашевич М. М.

Представлена методология разработки моделей машинного обучения для решения прикладных задач компьютерного зрения. В статье рассматриваются задачи компьютерного зрения, основные компоненты построения прикладных систем, проблемы и ограничения существующего технологического уровня.

Received 13.03.2024