Introductory Computer Vision Teaching Materials for VET Education

Vasja Lev Kirn,[1] Žiga Emeršič,[1] Gregor Hrastnik,[1] Nataša Meh Peer,[2] Peter Peer[1]

[1] University of Ljubljana, Faculty of Computer and Information Science, Ljubljana, Slovenia
vk3843@student.uni-lj.si, ziga.emersic@fri.uni-lj.si, gregor.hrastnik@scv.si, peter.peer@fri.uni-lj.si
[2] School center Velenje, Velenje, Slovenia
natasa.mehpeer@scv.si

Rapidly advancing development of artificial intelligence technologies, including deep learning techniques in the field of computer vision, has encouraged the need for early education about artificial intelligence in schools. This paper briefly describes the development of a computer vision curriculum, part of the AIM@VET (Artificial Intelligence Modules for Vocational Education and Training) EU project, targeting VET high-school students. The introductory materials presented in this paper are structured in three main teaching units (TUs), covering object detection and image segmentation. Each TU consists of eight tasks and a final assignment, totaling approximately 10 hours of classroom work. The course material, prepared in both traditional learning materials and in Python notebooks, combines theoretical concepts with practical coding exercises, with separate teacher and student versions. Materials rely on interactive tools and open-source libraries such as OpenCV, facilitating hands-on learning and immediate application of computer vision concepts.

DOI
https://doi.org/10.18690/um.feri.1.2024.5

ISBN
978-961-286-837-6

Keywords:
computer vision, Python notebook, object detection, teaching material, segmentation

University of Maribor Press

DOI
https://doi.org/
10.18690/um.feri.1.2024.5

ISBN
978-961-286-837-6

UVODNI UČNI MATERIALI RAČUNALNIŠKEGA VIDA ZA STROKOVNO SREDNJEŠOLSKO IZOBRAŽEVANJE

VASJA LEV KIRN,[1] ŽIGA EMERŠIČ,[1] GREGOR HRASTNIK,[1]
NATAŠA MEH PEER,[2] PETER PEER[1]

[1] Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Ljubljana, Slovenija
vk3843@student.uni-lj.si, ziga.emersic@fri.uni-lj.si, gregor.hrastnik@scv.si,
peter.peer@fri.uni-lj.si
[2] Šolski center Velenje, Velenje, Slovenija
natasa.mehpeer@scv.si

Ključne besede:
računalniški vid,
Python notebook,
detekcija objektov,
učna gradiva,
segmentacija

Hitro napredujoč razvoj tehnologij umetne inteligence, vključno s tehnikami globokega učenja na področju računalniškega vida, je spodbudil potrebo po zgodnjem izobraževanju o umetni inteligenci v šolah. Ta članek na kratko opisuje razvoj učnega načrta za računalniški vid, ki je del projekta EU AIM@VET (Artificial Intelligence Modules for Vocational Education and Training), namenjen srednješolcem poklicnega izobraževanja in usposabljanja. Uvodna gradiva, predstavljena v tem dokumentu, so strukturirana v treh glavnih učnih enotah, ki pokrivajo zaznavo objektov in segmentacijo slik. Vsak TU je sestavljen iz osmih nalog in zaključne naloge v skupnem okvirnem obsegu 10 ur izvajanja v učilnici. Gradivo je pripravljeno v tradicionalnih učnih gradivih in v zvezkih Python, združuje teoretične koncepte s praktičnimi vajami kodiranja z ločenimi različicami za učitelje in učence. Gradivo temelji na interaktivnih orodjih in odprtokodnih knjižnicah, kot je OpenCV, ki olajšajo praktično učenje in takojšnjo uporabo konceptov računalniškega vida.

Univerzitetna založba
Univerze v Mariboru

1 Introduction

The field of computer vision (CV) has rapidly evolved from its inception in the mid-20th century, undergoing significant advancements in the late 20th and early 21st centuries. Initially relying on handcrafted techniques, Computer Vision (CV) has witnessed a transformative leap with the introduction of deep learning methods, leading to fully automated and efficient approaches (IBM, 2023). CV now plays a crucial role in various real-world applications, from aiding medical image interpretation to enhancing surveillance systems and contributing to autonomous vehicles' success (Jahne, 2000, Janai, 2020). As digitalization becomes more prevalent, applications like consumer-grade facial detection are already integrated into daily life.

Recognizing the growing importance of CV education, this paper addresses the development and implementation of a CV course within the AIM@VET project, Aimed primarily at 16 to 20-year-old high-school students. Development was also a part of the diploma thesis (Kirm Lev, 2024). The CV course comprises of three modules, focusing on fundamental CV concepts through practical tasks, as already presented in (Emeršič, 2023). Here, we specifically present three teaching units (TUs) within the second module, covering Object Detection and Segmentation.

Key objectives of this work include responding to the need for CV education, targeting a specific age group, and developing TUs with tasks and assignments for effective learning. The coursework is presented digitally through Python notebooks, offering an interactive learning experience.

AIM@VET project, funded by the European Union aims to prepare the next generation for the increasing relevance of AI in the social and economic aspects of the future. It consists of the following groups and contents: (I) Spanish group: University of Coruña and Integrated Professional Training Center Rodolfo Ucha Piñeiro, focused on the robotics module, (II) Portuguese group: University of Minho and Caldas das Taipas Secondary School, focused on the ambient intelligence module, (III) Slovenian group: University of Ljubljana (UL) and School center Velenje (SCV), focused on the computer vision module.

1.1 Instructional Approaches to Computer Vision & Development

Traditional teaching methods in various fields, including computer science, often rely on textbooks for theoretical knowledge. However, practical understanding of programming and implementation, especially in areas like object detection and image segmentation, is essential in computer vision (CV). This course adopts an interactive approach using Python notebooks, consisting of eight tasks and a two-part assignment in each teaching unit (TU) to cover specific CV topics.

Introducing CV to high-school students is crucial due to its rapid growth, and providing hands-on experiences with visible outcomes, such as detecting edges in an image, can stimulate interest. The use of Python notebooks offers a visual and practical learning experience, enhancing student engagement and fostering further exploration in computer science (AIM@VET, 2023).

While CV is traditionally a higher-level topic in university studies, presenting it effectively can serve as an entry point for high-school students in their programming journey. Real-world applications like facial recognition make CV intriguing and relatable. The AIM@VET project aligns with modern teaching tools, utilizing Python Notebooks, like other online programming courses and initiatives like the AI+ project (AI Plus, 2023). AIM@VET builds on the insights gained from AI+, focusing on specialized AI education for secondary school students. The project continues the project-based learning approach with a hands-on emphasis and real-world tasks (Emeršič, 2023, AIM@VET, 2023).

2 Created Materials

This work encompasses the development of three Teaching Units (TUs), each consisting of 8 tasks and a final two-part assignment, totaling 30 hours of course content. Tasks, designed for both teachers and students as Python notebooks, progressively build on each other within a TU and logically connect between TUs, ensuring a comprehensive understanding of CV by the course's end.

Each task has two versions – one for students without exercise answers and less pre-written code, and one for teachers containing solutions. Corresponding course documents for each TU, initially in MS Word and later converted to LaTeX, provide a standardized structure for easy access. These documents outline the course plan,

background knowledge, and practical exercises, aiding teachers in guiding students through the material.

The first TU focuses on object detection fundamentals, covering basic image manipulation and introducing hand-crafted algorithms, laying the groundwork for further TUs, as shown in the exceprt from the materials, shown in Figure 1 and Fitgure 2. Figure one show a section from the PDF materials provided to the students, and Figure 2 shows an example of a Python code that students need to improve and run. Teaching unit 1 consist of two activities:

- Activity 1 Preparation for object detection: Learn to make simple object detection using templates, Bounding Box Techniques and Perform face detection, Image processing necessary for detection, Feature Extraction, and Image Pyramids.
- Activity 2 Object detection: Object Detection using Sliding Windows, Object Detection using Selective Search, Real-Life Examples of Object Detection, Evaluating Object Detection Algorithms, Assignment: Creating Simple Object Detection Algorithm

Figure 3: Bounding boxes around detected faces.

Required Knowledge for Task 3: Image Preprocessing for Object Detection: Image processing necessary for detection

As we have already learned in the past lessons of this course, Object Detection is a complicated task. We want to create an automatic detection of objects of interest within an image and also localize them, by drawing a bounding box around them. These two things need to be implemented well and work successfully in order for visual processing to be effective in real-world scenarios.

There are different techniques used in object detection, but before we input our image into our algorithm, we need to pre-process it in order to receive the best result.

Pre-processing an image for object detection means adjusting its properties in such a way that the objects we wish to detect and localize will be more easily detected and that the noise in our image (which results in false positives of detection) will be reduced. Techniques for pre-processing images include:

- Resizing the image
- Cropping a certain region of the image
- Applying color conversion on the image

Applying color conversion to an image

The first technique of pre-processing that we will explore is applying color conversion to an image. This means changing the relative spectrum of colors used to represent our image, which is then displayed as different values of colored pixels on our screen. In the example below we load the example image "Lenna"

Figure 1: Excerpt from the materials covering the required knowledge needed for the succcessful completion of the tasks in the TU.

Source: own

TU2 explores deep learning-based object detection, emphasizing modern approaches. It guides students through configuring and training the YOLOv5 deep learning model. TU2 consist of three activities, which contain:

- Activity 1 Introduction into deep learning: Introduction to Deep Learning-Based Object Detection, Popular Object Detection Models, Image processing necessary for detection, Setting up Environment and Dataset Preparation.
- Activity 2 Training detection model: Configuring the configuration files of YOLOv5 and understanding its architecture, Training the model using the custom dataset, Exploring the essential hyperparameters and optimizing them for improved model performance.
- Activity 3 Evaluating and improving detection: Testing the trained model on test image set and evaluating its performance, Assignment: Deep Learning Object Detection Algorithm.

TU3 shifts to image segmentation, covering fundamentals, techniques, and notable algorithms. It consists of three activities:

- Activity 1 Introduction into image segmentation: Introduction to image segmentation, Overview of image segmentation techniques, Image pre-processing for image segmentation.
- Activity 2 Implementing image segmentation algorithms: Image segmentation using thresholding, Image segmentation using region growing, Clustering for image segmentation, Image segmentation using watershed.
- Activity 3 Evaluating and improving segmentation: Evaluating image segmentation algorithms, Assignment: Image segmentation algorithms.

3 Evaluation and Adaptation of Course Materials

Teaching materials underwent adjustments during the creation of TUs 1, 2, and 3 for the second learning module. Feedback from a focus group and project coordination prompted structural changes.

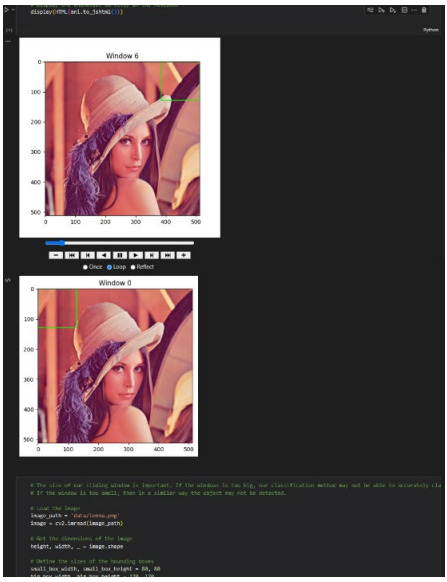

Figure 2: Teaching material example
Source: own

Initially, tasks in each Teaching Unit (TU) were outlined, adhering to the project's requirements. Ongoing collaboration with SCV allows for feedback and adjustments, presenting a snapshot of the current course version in this paper.

In response to initial feedback from SCV, the course structure was revised, materials simplified, TU shortened. The last two tasks were replaced by a comprehensive two-part assignment, fostering a more cohesive learning experience. Students sequentially complete tasks, building foundational knowledge, and then apply it independently in the final assignment, enhancing practical understanding.

The first adjustment involved creating two versions of each task: one for teachers with complete examples and solutions, and one for students with omitted solutions. This adaptation fosters critical thinking in students while aiding teachers in guiding them through tasks.

The second adjustment separated theoretical content from Python notebooks, relocating it to a standardized course document. This change facilitated a more collaborative classroom environment, allowing teachers to effectively convey theoretical knowledge.

The third adjustment reduced tasks in each TU from 10 to 8, with a final assignment covering two lessons. This shift aimed to focus on a smaller set of topics, ensuring a more cohesive learning experience.

Feedback from a focus group suggested simplifying content for high school students, leading to adjustments in task difficulty and the incorporation of simple exercises. Further feedback influenced changes to TU2, emphasizing step-by-step training of the YOLOv5 model, making the content more engaging and practical.

After materials were used at SCV, additional corrections were made to the materials, forming the final versions of the TUs. However, the ongoing development incorporates received feedback, aiming to create a more accessible and engaging course for both students and teachers.

4 Conclusion

This paper presents computer vision teaching materials, designed for students that are new to programming, and it introduces them fundamentals of AI techniques in computer vision. The course employs Python notebooks, emphasizing interactivity and engagement, allowing students to learn by doing and teachers to guide effectively.

Aligned with AIM@VET's goal of early AI education, this course targets young students, fostering interest through interactive learning. The adaptable structure facilitates future additions and revisions. Ongoing collaboration with SCV students ensures active adjustments, making the course responsive to student needs.

The development is ongoing until 2025, combining meaningful challenges and positive student feedback, promising a valuable resource for AI education. The course's expansion, coupled with AIM@VET's broader efforts, aims to equip students with essential AI knowledge, supporting their future studies and work.

References

IBM. "What is computer vision? " Pridobljeno iz: https://www.ibm.com/topics/computer-vision (2023).

Jahne, B., & Haußecker, H. "Computer vision and applications". Academic Press, 2000, pp. 609–660. (2000)

Janai, J, et al. "Computer vision for autonomous vehicles: Problems, datasets and state of the art". Foundations and Trends® in Computer Graphics and Vision 12.1–3 (2020), pp. 1–53. (2020)

Kirm Lev, V. "Teaching Units for a Computer Vision Course", Diplomsko delo, Pridobljeno iz: https://repozitorij.uni-lj.si/IzpisGradiva.php?id=153473. (2024).

Emeršič, Ž., Hrastnik, G, Meh, N., Peer, P. "Adapting VET Education to Labor Market Needs with Focus on Artificial Intelligence and Computer Vision", ROSUS 2023 (2023)

AIM@VET. "Guidelines". Pridobljeno iz: https://aim4vet.udc.es/backgroud/ (2023).

AI Plus. "AI Plus". Pridobljeno iz: https://aipudc.azurewebsites.net/ (2023).