

Introduction to non-contact temperature measurement procedures using the Python programming language

Predrag Stolić ^{1*}, Danijela Milošević ² and Zoran Stević ^{1,3}

¹ University of Belgrade-Technical Faculty in Bor, Bor, Republic of Serbia

² University of Kragujevac-Faculty of Technical Sciences Čačak, Čačak, Republic of Serbia

³ University of Belgrade-School of Electrical Engineering, Belgrade, Republic of Serbia

* pstolic@tfbor.bg.ac.rs

Abstract: *Learning temperature measurement procedures is extremely important for students of the technical professions. The previous practice was mostly based on the study of contact methods of measurement, however, due to the progress of technology, there is a need for a more detailed introduction to non-contact temperature measurement procedures, such as thermovision. When studying thermovision, in addition to the thermovision camera, the software supplied with it is also used to analyze the obtained thermograms. However, in most cases this software is not suitable for use in a teaching process. Due to this, the paper presents a solution based on the use of a different type of software compared to the traditionally used software during the introduction to thermovision procedures. The entire code for the software was developed in the Python programming language using elements of computer vision, image analysis and other capabilities that Python offers through its various modules. The entire environment intended for learning was implemented using virtualization techniques. An adequate virtual machine was implemented using free software and open source software.*

Keywords: *Engineering education; Image processing; OpenCV; Python; Thermovision*

1. INTRODUCTION

Temperature measurement is one of the most common measurements performed in process techniques [1]. The mentioned measurement procedure can be implemented in two ways: contact, when we make direct contact between the sensor and the measurement object, and non-contact, when there is no direct contact between the sensor and the measurement object [2]. Typical representatives of contact temperature measurement, in the field of process techniques, are resistance temperature sensors and thermocouples, while in the area of non-contact temperature measurement, optical and radiation (infrared) pyrometers, as well as thermovision (thermal imaging) cameras, are the most common.

Temperature measurements have important role not only in measurement systems, they are also important part of other systems such as control systems [3]. In many industrial processes in different systems, the contact type of temperature measurement still dominates, but it is expected in the future that in many places the contact type of measurement will be replaced by non-contact measurement, especially in monitoring systems. This statement is based on the fact that during contact measurement, the sensor does not make

contact only with the object it is measuring, but also with the rest of the measuring system in most cases. Also, there is a great influence of the environment, so during contact measurement, the sensor should be well protected, and special attention should also be paid to good thermal contact between the sensor and the object on which the measurement is performed.

In accordance with the above, future engineers will have to be prepared and trained to work, in addition to traditional methods of temperature measurement, with new measurement technologies such as non-contact measurement. That is why it is necessary to upgrade the appropriate curricula of the subjects that students of engineering study with parts related to the field of non-contact measurement, and of particular importance is the realization of both the theoretical part and the practical part.

This is the reason why in the new accreditation cycle (accreditation year 2020) for the subject Process measurement techniques, which is taken in the final year within the basic academic studies (undergraduate studies, level I of studies) of the Mining engineering study program at the Technical Faculty in Bor, there is a special part dedicated to studying the non-contact method of temperature measurement [4]. In addition to lectures, the

lessons that are realized within this course also include a certain number of hours of practical teaching. A certain number of these hours are intended for the realization of exercises in the field of non-contact measurement.

In the following, the implementation of practical exercises in the domain of using thermal imaging, as one of the methods of non-contact temperature measurement, will be shown, with a special emphasis on the implementation of the environment for performing the subject exercises, which was realized, among other things, by using free solutions, open source software and the Python programming language.

2. STRUCTURE OF EXERCISES AND EXPECTED OUTCOMES

Exercises in thermal imaging consist of three connected segments intended for realization in the faculty's interior and exterior space and part of the exercises is also realized through independent work at home. In the following, each of these segments will be briefly presented.

The first segment is realized in the external space of the faculty. It is intended for an introduction to the basic principles of thermovision and working with a thermovision camera. Recording thermograms of various objects, students learn to operate a thermovision camera adjusting the camera to the recording conditions by defining the emissivity for various recording objects, the distance of the recording object and the temperature of the environment in which thermal imaging tests are performed [5][6]. An example of one such recording is shown in Fig. 1.



Figure 1. Thermogram of a car wheel taken on a very sunny day

Thermograms made in this way are used later in the further work of students in the computer laboratory, as well as at home.

The second segment represents work with a thermovision camera in a computer laboratory. The camera is connected to the computer and continuous thermovision monitoring is performed over a specific observation object [7]. Also, at certain moments of time, thermograms are

extracted using a computer. Students learn continuous thermal imaging, as well as optimal indoor recording conditions by monitoring the temperature and relative humidity in the room with adequate equipment (suitable probes) and adjusting the camera to the given conditions [8][9]. Thermograms made in this way, also, are used later in the further work of students.

The third segment is a final part of thermovision exercises which is partly realized in the computer laboratory and the rest of activities are realized according to the principle of working from home. In this part, students are introduced to the analysis of thermograms by processing recorded thermograms obtained through previously realized activities.

3. SOME POTENTIAL SOFTWARE ISSUES

As can be seen from the above, only in the first part the exercises are performed exclusively using a thermovision camera, while in the other parts the exercises involve the use of a computer. This practically implies the use of adequate software.

All manufacturers of thermovision cameras deliver appropriate software along with cameras, however, there are some issues related to the usability of such software in teaching from several aspects. Most of such software are intended for professional use and are not adequate for use in the teaching process. Also, since it is commercial software, its use in teaching is limited by various license conditions. There are limited possibilities regarding the redistribution of the software, which practically means that the software cannot be made available to students at all times, especially not for working from home. Certain softwares have limitations in the number of users, so they cannot be installed on a sufficient number of computers intended for performing exercises.

It should also be noted that most of the software is not multi-platform oriented, so it may happen that it is only partially usable in the implementation of exercises. The situation in which the hardware is controlled by the software is not desirable, but it is necessary to enable the operation of the software regardless of the type of platform on which it is intended to be launched, or at least to make it feasible in most cases.

Certain software that comes with a thermovision cameras also requires an extensive registration process before using the software itself, which can cause a number of problems from the Law on Personal Data Protection point of view. In some cases, certain functionalities of the software are limited in some way (limited by the number of uses, locked in their entirety, etc.) until the software is upgraded, which in most cases is charged additionally.

4. DEVELOPMENT OF AN ALTERNATIVE SOLUTION

Due to the reasons mentioned above, it was decided that the exercises are not carried out using the software supplied with the thermovision camera, but that the exercises are carried out through alternative solution, which will be presented in the following lines.

4.1. Environment

Students use various operating systems on their computers at home. Also in the computer laboratory, although the aim is for all computers to be identical in terms of platform, certain deviations may occur, so we may have a certain diversity in terms of the version of the operating system, versions of certain software, configurations and the like.

The goal is to ensure that the software used for the implementation of the exercises can work identically on Linux, Windows 7/8/8.1/10/11 and macOS systems, because each of these systems can be encountered in real work with students. Of course, it is possible to compile the software separately for each of these systems, but in this case it would not be the most adequate solution.

If the software were to be compiled separately for each of these systems, the focus is again on the student to choose the appropriate installation and to configure it on his computer, as well as to eliminate any errors that may occur during the installation. Also, here again the problem of meeting the minimum requirements for running the software, which can vary significantly even between similar systems, is manifested. For example, compiling software for Windows 10 can be done using the Python 3.9 compiler, but it cannot be used for Windows 7, since the last version of the Python compiler that enables compiling for Windows 7 is 3.7, so the work of compiling for Windows systems becomes even more complicated.

That is why it was decided not to compile the software, but to provide the student with an environment in which the Python code will be run and that the environment be identical for everyone, regardless of the platform used. Anaconda Distribution was chosen to run all the necessary Python programs, since it is an open source platform that integrates everything that is necessary for a student to successfully master the exercises. It integrates the appropriate Python compiler, the necessary libraries, as well as the Jupyter Notebook IDE, and in addition, it offers simplicity of implementation and ease of management, which is important when working in this special case when the focus is on mastering the course material by the student.

However, the installation of Anaconda itself, setting up the environment, installation of all the necessary

modules and similar activities can again have a certain amount of complexity for the average student. At the same time, it should be kept in mind that for Mining engineering students focus is not on computer science, but software is an auxiliary tool that contributes to better mastering of professional subjects.

Analyzing the above, it is concluded that students should be provided with a pre-installed environment that the student will only run without any additional installations, settings, etc. Accordingly, the concept of virtualization was introduced into the entire solution, since it shifts the focus of the environment, installation and configuration of the software from the student to the teacher who will prepare and make available that virtual environment.

In accordance with the above requirements, a virtual machine was created that the students will use both in the computer lab and on their computers at home in order to have a pre-defined environment that will be completely identical on all platforms and all computers, regardless of whether it is about faculty computers or personal computers at home.

For the guest OS, Linux Fedora 36 in the Workstation edition was selected, within which the Anaconda Distribution was installed along with all the necessary modules that are necessary to run the software. Linux Fedora was chosen for this task because it can be completely freely used, redistributed and there are no additional limitations. The entire environment is fully configured and tested, so that the student only has to run the specific Python software and realize his intended teaching activities. The student does not have the obligation to download the necessary Python files, since they, together with the entire teaching material, are integrated into the existing virtual machine so that the student does not have to make additional efforts in finding the necessary material.

The operation of the virtual machine is made possible by using the VirtualBox product of the company Oracle. VirtualBox was chosen for the reason that it is implemented as a type 2 hypervisor, which practically means that it runs on the user's operating system and then enables the running of guest OSES on virtual machines using basic virtualization techniques. In addition, VirtualBox exists in appropriate versions for Linux, Windows and macOS systems, it is very easy to implement and to use and is completely free and is an open source software that can be freely redistributed further.

In this way, a functional and complete solution was created that can be launched on any platform in a simple way. At the same time, all elements are always identical: the way of use, display implementation, configuration, paths, files and all

other elements always implement identical properties regardless of the platform on which the virtual machine is launched, which was aimed at from the very beginning of the planning of the subject exercises.

4.2. Code

As already mentioned, the entire code was implemented using the Anaconda Distribution environment. In this case, Jupyter Lab and Jupyter Notebook were equally used as the IDE in which the given code was implemented (that's why the codes have the .ipynb extension instead of the traditional .py extension). The entire code was developed using the basics of the Python3 programming language, and the compiled code itself was compiled using the Python 3.9 compiler.

Two codes were developed for the purposes of mastering the field of thermovision, which is studied within the subject course:

TCam.ipynb for connecting the thermovision camera to the computer and continuous monitoring of the observed object through the thermovision camera and

TTemp.ipynb for computer analysis of thermograms recorded by a thermovision camera.

When implementing the first code TCam, the OpenCV library for Python was used in version 4.6.0, which contains the appropriate functionalities related to the use of computer vision [10]. The connection between the thermovision camera and the computer is achieved through a USB connection, so in essence the access to the thermovision camera will not differ significantly from the access to any other type of camera using OpenCV. Of course, a thermovision camera, unlike others, will emit a thermographic image (thermogram), but it will do so frame by frame (each frame will be one thermogram) as other cameras do. This means that the access to the thermovision camera, with some slight modifications, can be done entirely based on the code that is an integral part of the OpenCV documentation related to the Python programming language [11]:

```
#Capturing signal from thermovision camera
import cv2

video = cv2.VideoCapture(0)

if not video.isOpened():
    print("Nije ostvarena konekcija sa kamerom")
    exit()

while True:
    ret, frame = video.read()

    if not ret:
```

```
        print("Greska u prenosu sa kamere")
        break

    cv.imshow('TCam', frame)

    if cv.waitKey(1) == ord('q'):
        break

video.release()
cv.destroyAllWindows()
```

The example of the captured video signal from the thermovision camera by starting the previous code using Jupyter Notebook is shown in Fig. 2.

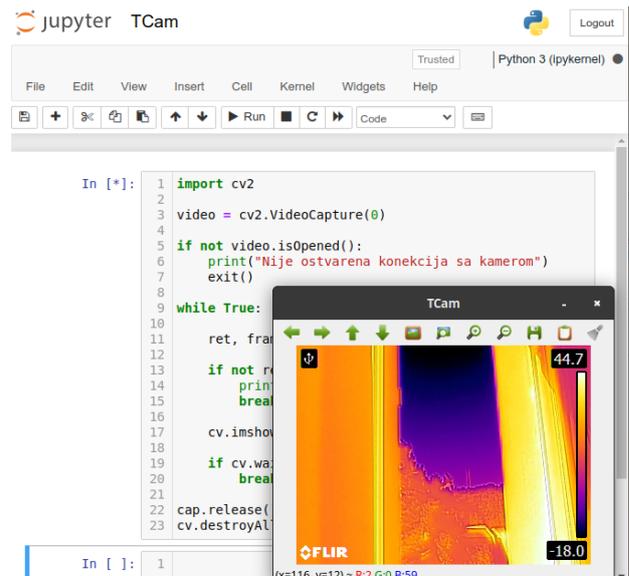


Figure 2. TCam.ipynb is running in Jupyter Notebook and captures the video signal from thermovision camera

Each received frame can be saved as a corresponding file on disk. It has already been mentioned that each frame represents one thermogram and in accordance with the previous one, based on the captured video signal from the thermovision camera, the corresponding thermogram can be recorded on the computer disk. Second code TTemp is a tool for analyzing the obtained thermograms. It can work in three defined ways:

- it can find the maximum temperature recorded on the thermogram and mark the points where the maximum temperature was detected,
- it can find the minimum temperature recorded on the thermogram and mark the points where the minimum temperature was detected and
- for the temperature value provided by the user, it can find all appearances of that temperature on the thermogram and mark appropriate points where the temperature was detected.

The thermogram analysis in this code is based on the use of the Pillow library, which represents a PIL (Python Imaging Library) fork, while the GUI

implementation is based on the use of the Tkinter library, which enables a clear graphic interpretation of the obtained processing results [12].

Regardless of the method of analysis of the existing thermogram, the first step that must be taken is the translation of the given temperature scale [13]. The temperature scale is given in the right part of the thermogram, and on it, the corresponding temperature values are represented by appropriate color representations. The minimum temperature value on the scale is marked at the bottom of the scale, while the maximum value on the scale is marked at the top of the scale. Knowing this data, we can find the representation of other temperature values on the scale by using the following code:

```
def UcitajSkalu():
    global ScalePixel, ScaleTemperature
    Y1=30
    Y2=209
    X=310
    Step=(High-Low)/(Y2-Y1)
    Temperature=High
    ScalePixel=[]
    ScaleTemperature=[]
    for i in range(Y1,Y2):
        ScalePixel.append(loaded_image.getpixel((X,i)))
        ScaleTemperature.append(Temperature)
        Temperature=Temperature-Step
```

X and Y values are determined by the characteristics of the thermovision camera itself. In this case, a FLIR E5 thermovision camera was used, which records thermograms with a resolution of 320x240 pixels. Thus, it was determined that on each thermogram recorded by this camera, the temperature scale is located on 310th pixel on the x axis and from the 30th to the 209th pixel along the y axis.

When you know which color coded which temperature value, you can easily continue with the analysis of the thermogram. The thermogram is analyzed in two passes. In the first pass, the thermogram is read pixel by pixel and to each pixel is assigned a corresponding temperature value. In the second pass, depending on the selected option, the maximum recorded temperature value is found and all positions where these values were detected are marked, or the minimum recorded temperature value is found and all positions where these values were detected are marked, or all positions are detected in which the temperature defined by the user himself, i.e. the student, is recorded. Standard algorithms are used for mutual comparison of values and finding extremes.

At the end of each thermogram processing, the student can record a new thermogram with the

corresponding results of processing. Saving process using the filedialog option, while the new thermogram itself is recorded as an image given in JPEG format. An example of a recorded thermogram with marked positions where maximum temperature values were identified is given in Fig. 3 (the rear window of the parked car is analyzed).

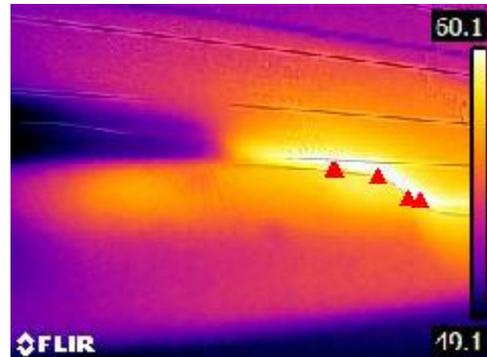


Figure 3. Example of saved thermogram with obtained results after detection of spots with maximum temperature recorded

It should be noted that at any moment, if any unwanted behavior of the software occurs during thermogram analysis, the student has the option of resetting. The reset option deletes all detected values, re-initializes all parameters and loads the last good known thermogram, so that the student can start again with the analysis of the desired thermogram in order to obtain adequate processing results.

5. CONCLUSION

As can be seen from the above, it is entirely possible to teach students about the basics of non-contact temperature measurement by using some new non-traditional thermal imaging solutions implemented using the Python programming language and its corresponding libraries. In addition, the presented solution frees the student from some secondary activities that he would have to perform using traditional software, such as software installation, system configuration, troubleshooting, which can be an extremely tiring process, especially for students who do not pursue their education in the field of computer science. Also, by uniting all teaching materials, examples, software and other digital contents within one virtual machine, one centralization of the necessary contents is realized with the aim of reducing the time needed to find those contents through various sources.

Various benefits are achieved by implementing the presented approach in the teaching process.

If we look at the financial aspect, it can be said that the implementation can be achieved according to the "zero-cost" principle. The entire software implies the use of free solutions, which practically means that there are no additional investments in

terms of software by the institution that implements the teaching process. There is also no investment by the student side either, since the mentioned virtual machine can be used freely without any compensation, that is, there are no additional licensing costs.

Also, if we look at the legal regulations, it should be noted that there are certain license provisions even when using free software and open source software. However, these licenses are much more flexible than most licenses practiced by various software manufacturers and suppliers. Most of the licenses that are practiced when licensing free software support the free redistribution, which is of great importance when incorporating the software into the teaching process, since the software is installed outside the borders of the institution and on equipment that is the personal property of students. The selection of software that is part of the virtual machine is realized in such a way that it does not require any additional registration or the delivery of any sensitive data to a third party, which may cause certain problems regarding the right to the protection of personal data in accordance with applicable laws. In this way, legal protection was realized at different levels, the institution is protected, the students are protected and the teaching staff are also protected from potentially controversial situations.

From the sustainability point of view, it can be said that significant results are being achieved here as well. The presented environment with all its elements was primarily implemented for the purposes of teaching within the Process Measurement Techniques course, which is taken by students of Mining Engineering, but the use value exceeds these limits. It can be used in the scientific research work of students, which in recent years has been increasingly emphasized in the system of higher education and also it can be useful to students of other technical fields, for example it can realize a very large potential in the implementation of the teaching process in computer science subjects. Through described approach, students can learn the basics of programming in the Python programming language, learn the basics of virtualization, learn the basics of connecting a computer to other devices within measuring systems, gain insight into advanced programming, computer vision, designing and implementing a graphical user interface (GUI), etc. The possibilities are numerous and with slight modifications this solution can realize many other activities in different teaching and learning processes.

The complete learning model described in the paper will be used regularly in the teaching process of the Process Measurement Technique course during the spring semester of the 2022/2023 school year. It is expected that after the above mentioned semester, a proper evaluation will be carried out, and if the

results are positive, some other solutions will be implemented in the teaching process of other subjects based on the presented model.

REFERENCES

- [1] Popović, M. (2004). *Senzori i merenja*. Srpsko Sarajevo, Republic of Srpska, Bosnia and Herzegovina: Zavod za udžbenike i nastavna sredstva.
- [2] Norton, H. (1989). *Handbook of Transducers*. Englewood Cliffs, New Jersey, USA: Prentice-Hall, Inc.
- [3] Randić, S. (2010). *Inteligentni senzori*. Čačak, Republic of Serbia: University in Kragujevac, Faculty of Technical Sciences Čačak.
- [4] University of Belgrade-Technical faculty in Bor, Book of courses for study program Mining engineering - undergraduate studies (on Serbian). (2019). Available at https://www.tfbor.bg.ac.rs/files/doc/studijski-programi/RI/2019/oas_ri_knjiga_predmeta.pdf Last accessed on July 5, 2022.
- [5] Thermography Pocket Guide. (2017). Lenzkirch, Germany: Testo SE & Co. KGaA
- [6] Introduction to Thermography Principles. (2009). Orlando Park, Illinois, USA: American Technical Publishers Inc., Fluke Corporation and The Snell Group.
- [7] Stević, Z., Rajčić-Vujasinović, M., Antić, D. (2008). *Primena termovizije*. Bor, Republic of Serbia: University of Belgrade, Technical Faculty in Bor.
- [8] The Complete Guidebook on Thermal Screening for Elevated Skin Temperature. (2020). Wilsonville, Oregon, USA: FLIR Systems, Inc.
- [9] Stolić P., Stević Z., Stanimirović Z., Stanimirović I. (2022). Implementation of anti-covid measures in university educational process taking advantage of thermal imaging approach. In: *Proceedings of the XXIII international scientific-practical conference Modern information and electronic technologies ("MIET-2022")*. Odesa, Ukraine, 34-37
- [10] Howse J., Minichino J. (2020). *Learning OpenCV 4 Computer Vision with Python 3*. 3rd ed. Birmingham, UK: Packt Publishing
- [11] OpenCV Documentation. Getting Started with Videos. Available at https://docs.opencv.org/4.x/dd/d43/tutorial_py_video_display.html Last accessed on July 8, 2022
- [12] Moore A. (2018). *Python GUI Programming with Tkinter*. Birmingham, UK: Packt Publishing
- [13] Stolić, P., Peulić, A., Tanikić, D. (2018). Software development for thermovision application in triage procedures of emergency conditions. In: *Proceedings of the 26th International Conference on Ecological Truth and Environmental Research - EcoTER '18*, Bor Lake, Serbia, 379-384.