A Qt-Based Image Processing Application

MANOLOTH Ketsavanh
Thai Nguyen University of Information and Communication Technology
Thai Nguyen University, Thai Nguyen, Vietnam
student554@ictu.edu.vn

***Abstract*—This undergraduate thesis investigates an image processing application built with Qt tools and Python. A desktop image tool can help learners inspect effects such as grayscale conversion, blurring, and face detection. The student studies image-processing algorithms and uses Qt Designer, PyQt, Python, and OpenCV to construct an interface. The thesis documents design diagrams, programming steps, and example image-processing screens. The source shows an implemented application with image display and processing functions. It includes code or descriptions for grayscale and blur effects and face detection or recognition. It does not report a controlled recognition benchmark or broad end-user trial. The manuscript emphasizes implementation and illustration. Recognition behavior may change with image quality and camera conditions, and the source does not establish independent accuracy or reliability across varied subjects. The contribution is a documented student study or prototype whose stated scope can guide further testing.**

***Index Terms*—Qt, PyQt, OpenCV, image processing**

I. Introduction

A desktop image tool can help learners inspect effects such as grayscale conversion, blurring, and face detection. This thesis investigates an image processing application built with Qt tools and Python. It begins from a practical engineering need and explains the technologies and workflow chosen for the case. The objective is to document a student implementation or comparative investigation, together with the evidence available in the manuscript. Broader operational or economic effects require separate measurements and are not assumed from the existence of a prototype [1]. Earlier ICTU student work on related designs is included here for context [2]. Related published studies provide context for the subject and methods [3], [4], [5], [6], [7], [8], [9], [10]. These contextual works were selected for this English summary; they are not asserted to appear in the original thesis bibliography.

II. Methods

The student studies image-processing algorithms and uses Qt Designer, PyQt, Python, and OpenCV to construct an interface. The thesis documents design diagrams, programming steps, and example image-processing screens. The source is summarized through its stated design, implementation material, examples, and conclusion. Features shown in the manuscript are distinguished from proposals for later development. This review does not add a new experiment, fill missing measurements with estimates, or treat a planned capability as an observed result. The source thesis remains the provenance for the reported work [1].

Supervisor: Nguyễn Thị Bích Điệp, Thai Nguyen University of Information and Communication Technology (ntbdiep@ictu.edu.vn).

III. Results

The source shows an implemented application with image display and processing functions. It includes code or descriptions for grayscale and blur effects and face detection or recognition. It does not report a controlled recognition benchmark or broad end-user trial. The report's screens, configurations, examples, or measurements constitute the evidence for this summary. A capability described only as a design intention is not counted as a validated operational outcome. No additional performance, adoption, financial, or safety figure has been introduced here. The result is interpreted within the scope of the student's documented setup [1].

IV. Discussion

The manuscript emphasizes implementation and illustration. Recognition behavior may change with image quality and camera conditions, and the source does not establish independent accuracy or reliability across varied subjects. Later work could separate each function into repeatable tests, evaluate processing time and recognition errors, and improve interface accessibility. These limitations affect how the evidence should be read. The current manuscript can support understanding of the design and reported demonstration, while independent tests would be needed for stronger claims. Any later deployment should document its environment and separate observed outcomes from intended benefits [1].

V. Conclusion

The thesis documents an image processing application built with Qt tools and Python. The source shows an implemented application with image display and processing functions. It includes code or descriptions for grayscale and blur effects and face detection or recognition. It does not report a controlled recognition benchmark or broad end-user trial. The manuscript emphasizes implementation and illustration. Recognition behavior may change with image quality and camera conditions, and the source does not establish independent accuracy or reliability across varied subjects. Later work could separate each function into repeatable tests, evaluate processing time and recognition errors, and improve interface accessibility. Accordingly, the present contribution is a source-grounded account of the student work and its tested or illustrated scope. Claims about broader effectiveness should be assessed in later work using explicit evaluation conditions [1].

References

[1] MANOLOTH Ketsavanh, ”A Qt-Based Image Processing Application,” internal undergraduate thesis (original in Vietnamese), Thai Nguyen University of Information and Communication Technology, 2024.

[2] Nguyễn Ngọc Nguyên Giáp, ”BUILDING AN IMAGE PROCESSING APPLICATION WITH JAVA,” internal undergraduate thesis, Thai Nguyen University of Information and Communication Technology, 2021.

[3] Eiichiro MOMMA, ”Image Processing with Python and OpenCV for Beginners,” Journal of the Japan Society for Precision Engineering, 2020. doi:10.2493/jjspe.86.761.

[4] Mrs. Asha K H et al., ”Python - Based Image Processing,” International Journal of Scientific Research and Management, 2021. doi:10.18535/ijsrm/v9i11.ec03.

[5] Ujjwal Sharma , Tanya Goel , Dr. Jagbeer Singh, ”Real-Time Image Processing Using Deep Learning With Opencv And Python,” Journal of Pharmaceutical Negative Results, 2023. doi:10.47750/pnr.2023.14.03.246.

[6] Tsuyoshi Suenaga et al., ”OpenCV 2.0 (3); introduction of various image processing functions and face detection,” The Journal of The Institute of Image Information and Television Engineers, 2010. doi:10.3169/itej.64.1362.

[7] Shun Jian Yin et al., ”Application of Machine Vision in ARM Based on OpenCV and Qt,” Advanced Materials Research, 2014. doi:10.4028/www.scientific.net/amr.1049-1050.1938.

[8] Zhen Feng Huang et al., ”The Study of Image Processing Method Based on OpenCV and USB Industrial Camera,” Applied Mechanics and Materials, 2012. doi:10.4028/www.scientific.net/amm.220-223.2721.

[9] Aji Bijaksana Abadi et al., ”Digital Image Processing for Height Measurement Application Based on Python OpenCV and Regression Analysis,” JOIV : International Journal on Informatics Visualization, 2022. doi:10.30630/joiv.6.4.1013.

[10] Ferenc Vajda, ”Application and implementation of window-based image processing algorithms,” Microprocessing and Microprogramming, 1990. doi:10.1016/0165-6074(90)90281-d.