# CHƯƠNG IV. TỔNG KẾT

## Kết quả đạt được

Báo cáo này đã đạt được những kết quả quan trọng trong việc nghiên cứu và ứng dụng thuật toán K-means vào xử lý ảnh, đồng thời cung cấp nền tảng lý thuyết và kỹ thuật cần thiết để thực hiện dự án. Các kết quả chính đạt được bao gồm:

* Hiểu biết về Python và các thư viện hỗ trợ: Đã trình bày khái quát về ngôn ngữ Python, các thư viện quan trọng như Numpy, Sklearn, Scipy, và Matplotlib. Những thư viện này là nền tảng cho việc triển khai thuật toán và xử lý dữ liệu, hình ảnh một cách hiệu quả.
* Nghiên cứu về hệ màu và lưu trữ dữ liệu: Hiểu rõ hệ màu và các cách lưu trữ dữ liệu ảnh giúp tối ưu hóa việc sử dụng tài nguyên, chuẩn bị tốt cho quá trình phân cụm và xử lý ảnh.
* Nắm bắt lý thuyết K-means và các ứng dụng trong xử lý ảnh: Đã phân tích kỹ thuật hoạt động của K-means, giải thích các thuật ngữ, ý nghĩa và ứng dụng thực tế của thuật toán này, đặc biệt trong việc phân cụm màu sắc cho ảnh.
* Ứng dụng K-means trong xử lý ảnh: Đã triển khai và thử nghiệm thuật toán K-means cho phân cụm màu ảnh, với kết quả đạt được là hình ảnh đơn giản hóa nhưng vẫn giữ được các vùng màu chính. Điều này chứng minh hiệu quả của thuật toán trong việc nén và phân tích hình ảnh.

Báo cáo đã hoàn thành mục tiêu đặt ra, cung cấp kiến thức nền tảng và minh họa rõ ràng hiệu quả của thuật toán K-means trong xử lý và phân tích hình ảnh

## Hạn chế

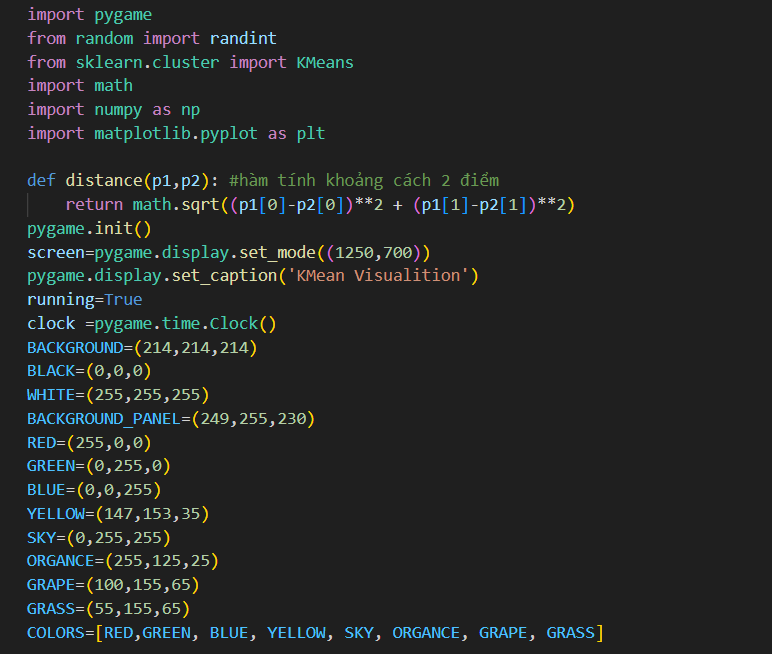
Dù báo cáo đã đạt được nhiều kết quả tích cực, nhưng vẫn còn một số hạn chế cần lưu ý:

* Giới hạn trong việc lựa chọn số cụm (k): Thuật toán K-means yêu cầu người dùng xác định trước số cụm, điều này có thể dẫn đến kết quả phân cụm không tối ưu nếu lựa chọn số cụm không chính xác. Điều này đặc biệt khó khăn khi làm việc với các hình ảnh phức tạp có nhiều màu sắc.
* Phụ thuộc vào vị trí khởi tạo: K-means có thể cho kết quả khác nhau tùy thuộc vào cách chọn các tâm cụm ban đầu. Việc khởi tạo không chính xác có thể khiến thuật toán bị kẹt trong cực trị cục bộ, làm giảm độ chính xác của kết quả phân cụm.
* Không hiệu quả với các cụm có hình dạng phức tạp: K-means hoạt động tốt nhất khi các cụm có dạng hình cầu và có kích thước tương đồng. Đối với các hình ảnh có cấu trúc phức tạp hoặc các cụm có hình dạng phi tuyến tính, thuật toán có thể không nhận diện đúng các cụm màu.
* Độ nhạy với nhiễu: K-means dễ bị ảnh hưởng bởi các điểm ảnh nhiễu hoặc các điểm ảnh ngoại lệ (outliers), dẫn đến việc phân cụm không chính xác hoặc mất đi màu sắc chủ đạo trong ảnh.
* Tính toán tốn kém cho dữ liệu lớn: Với các hình ảnh có độ phân giải cao, số lượng điểm ảnh rất lớn, làm tăng khối lượng tính toán và thời gian xử lý của thuật toán. Điều này có thể ảnh hưởng đến hiệu suất, đặc biệt khi áp dụng cho những dự án xử lý ảnh quy mô lớn.
* Khả năng phân cụm hạn chế trong xử lý ảnh phức tạp: Trong một số hình ảnh phức tạp, việc phân cụm chỉ dựa vào màu sắc có thể không đủ để tách biệt các đối tượng mong muốn, dẫn đến việc phân vùng không chính xác giữa các đối tượng trong ảnh.

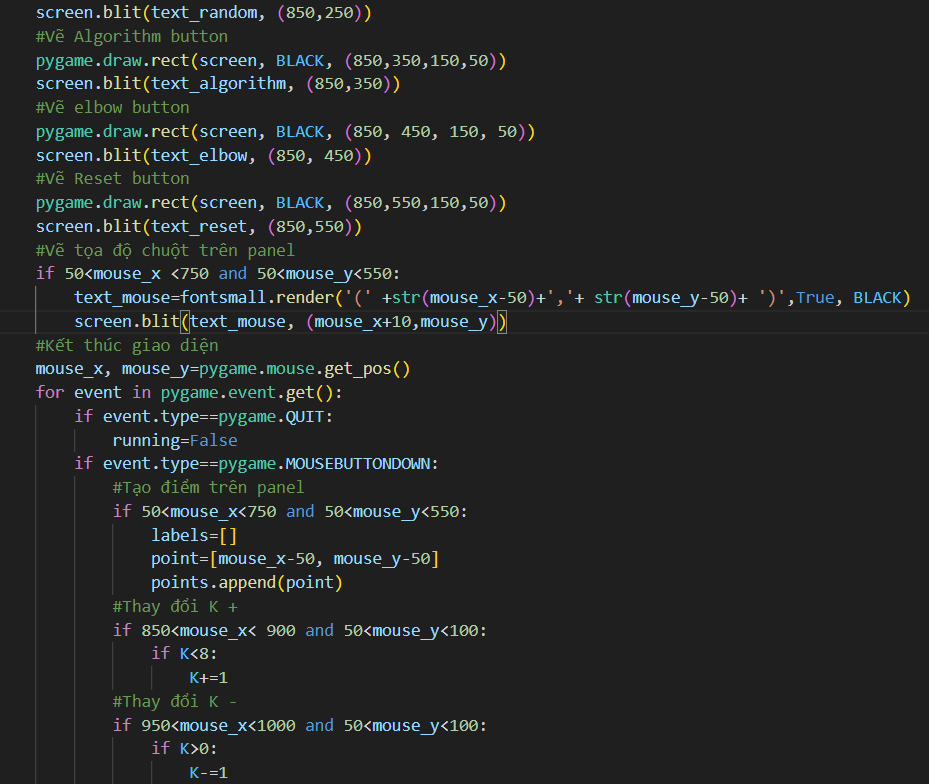
Những hạn chế này có thể được cải thiện trong các nghiên cứu tiếp theo, như thử nghiệm các thuật toán phân cụm khác hoặc tối ưu hóa quy trình khởi tạo và xử lý nhiễu.

**PHỤ LỤC**

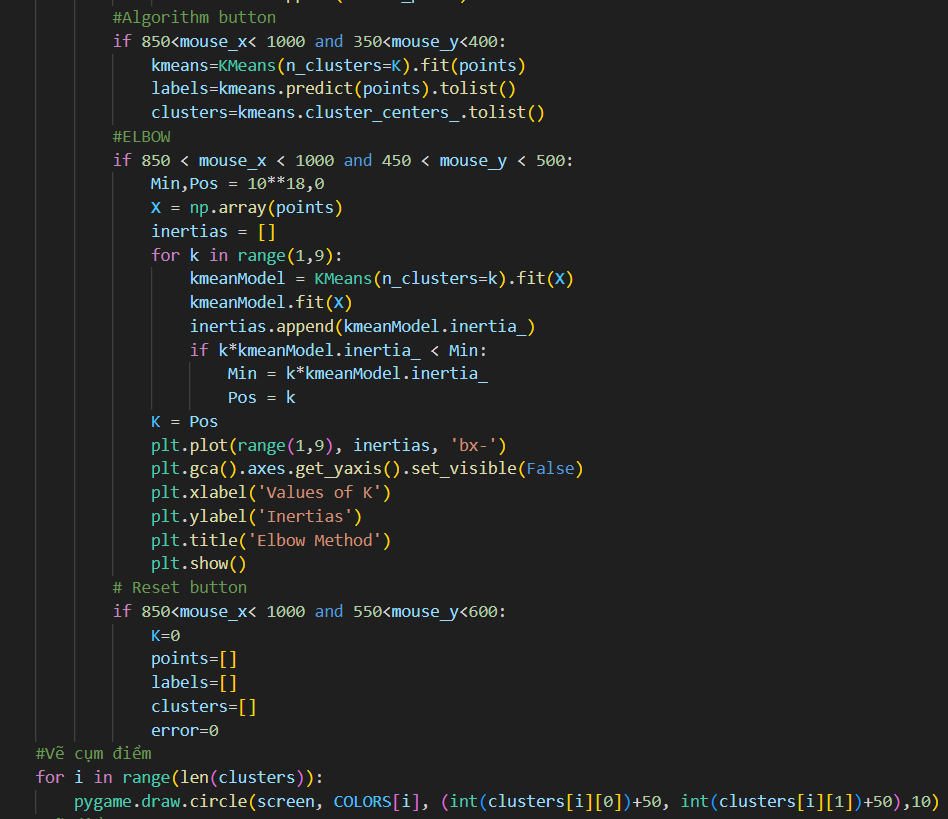
* 1. Mô phỏng thuật toán K-means bằng ngôn ngữ python:

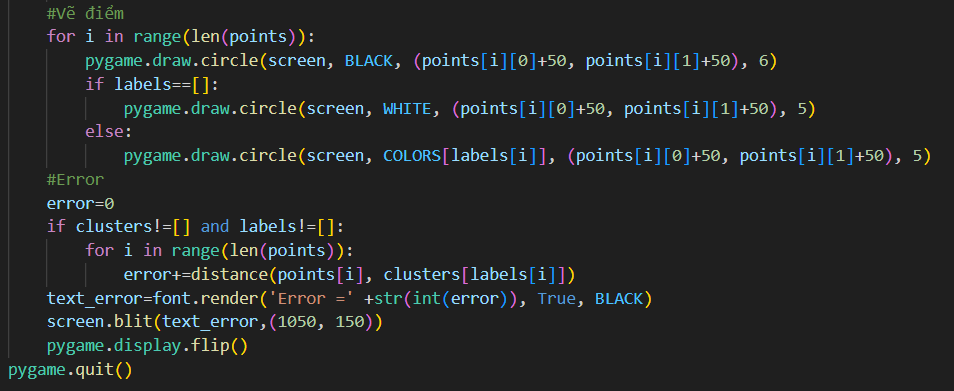




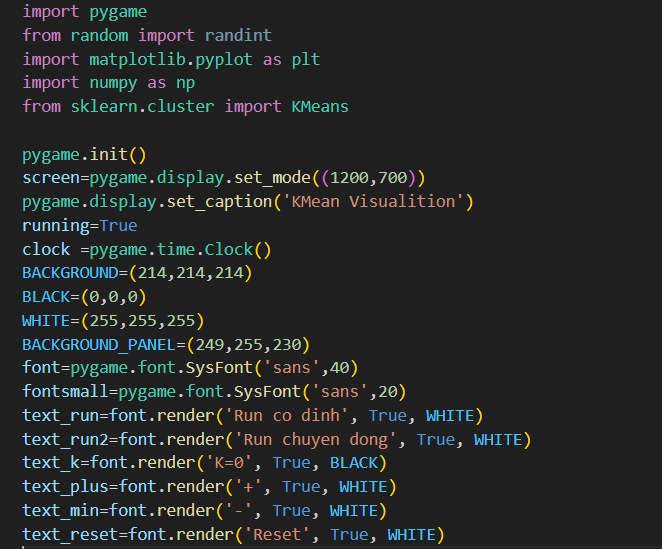


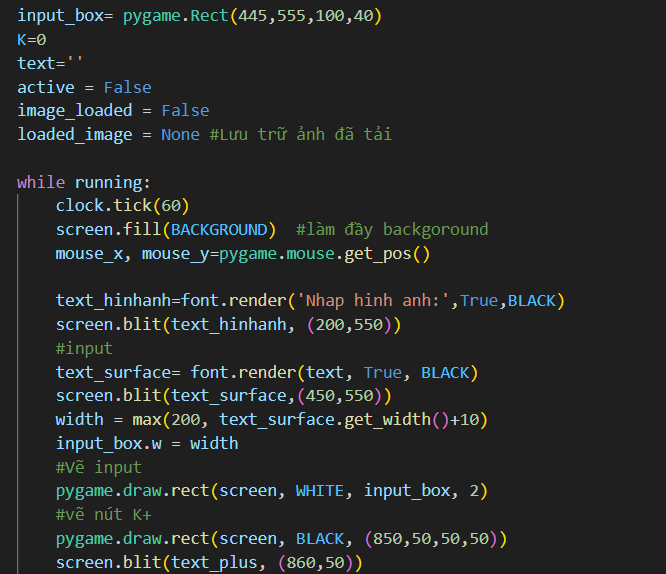


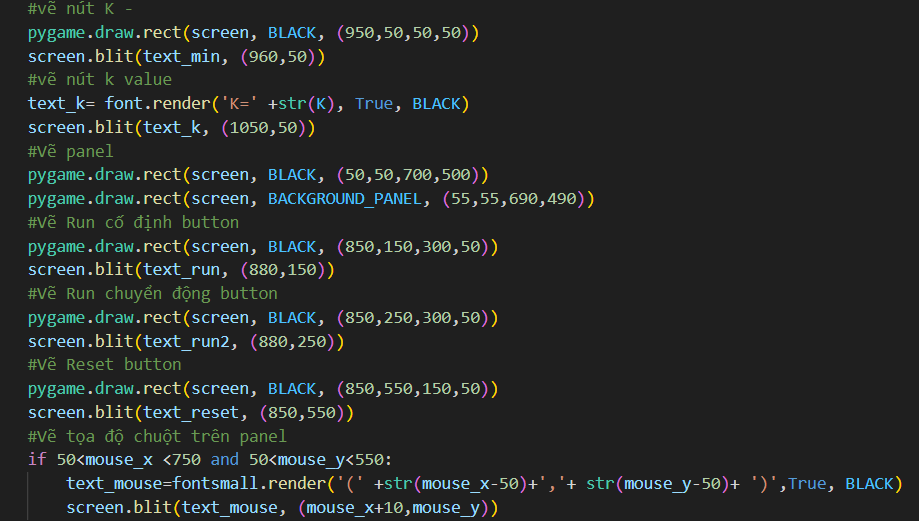


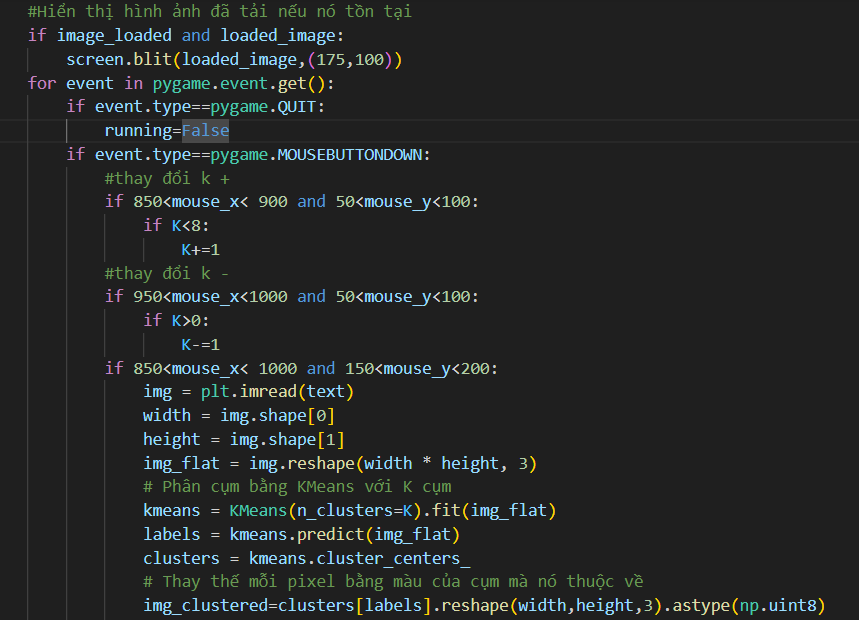


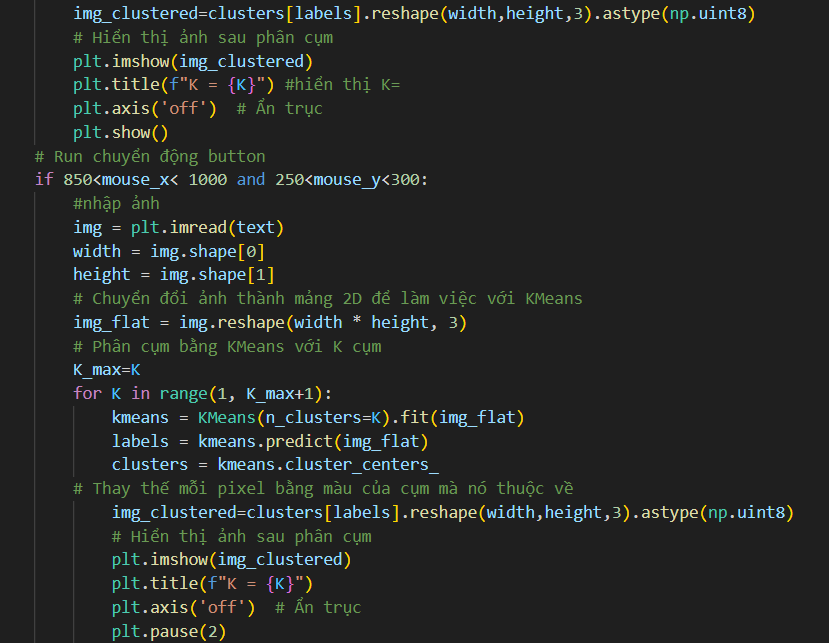
* 1. Xử lý ảnh bằng thuật toán Kmeans:

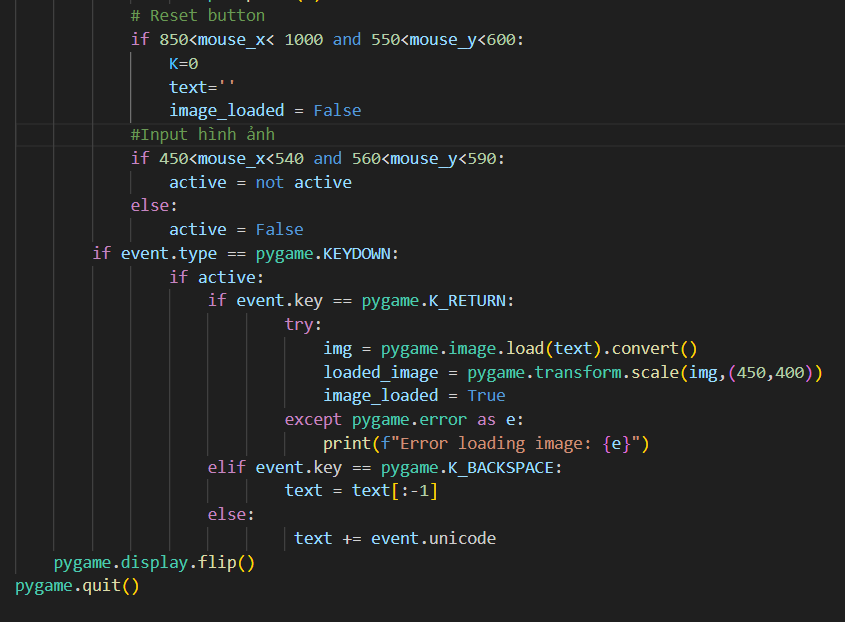












# TÀI LIỆU THAM KHẢO

1. Aggarwal, C. C. (2013). *Data mining: Concepts and techniques*. Springer.
2. Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.
3. Bivand, R. S., Pebesma, E., & Gomez-Rubio, V. (2013). *Applied spatial data analysis with R* (2nd ed.). Springer.
4. Bradski, G., & Kaehler, A. (2008). *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media.
5. Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern classification* (2nd ed.). Wiley.
6. Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems* (2nd ed.). O'Reilly Media.
7. Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2nd ed.). Springer.
8. Hunter, J. D. (2007). *Matplotlib: A 2D graphics environment*. *Computing in Science & Engineering, 9*(3), 90–95
9. Lloyd, S. (1982). *Least squares quantization in PCM*. *IEEE Transactions on Information Theory, 28*(2), 129–137.
10. McKinney, W. (2010). *Data structures for statistical computing in Python*. *Proceedings of the 9th Python in Science Conference*, 51–56.
11. Murphy, K. P. (2012). *Machine learning: A probabilistic perspective*. MIT Press.
12. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). *Scikit-learn: Machine learning in Python*. *Journal of Machine Learning Research, 12*, 2825–2830.
13. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Fei-Fei, L. (2015). *ImageNet large scale visual recognition challenge*. *International Journal of Computer Vision, 115*(3), 211–252.
14. https://stackoverflow.com/search?q=kmeans&s=50b5555f-a20f-414e-86eb-1b2e90c53bbb