Automatic Image Classification with AutoImageClassifier

Igor Tomkowicz
Faculty of Electronics, Telecommunications and Informatics
Gdańsk University of Technology
Gdańsk, Poland
s194103@student.pg.edu.pl

***Abstract*—This paper presents "AutoImageClassifier," a system for automatic image categorization, addressing the challenge of efficiently managing large and diverse image collections. It employs transfer learning with MobileNetV2 for robust feature extraction, hierarchical agglomerative clustering for insightful unsupervised grouping, and k-Nearest Neighbors (k-NN) for effective supervised classification into predefined categories. The project demonstrates a complete, configurable pipeline, enabling a comparative analysis of both learning paradigms and offering a practical framework for organizing potentially unlabeled image datasets, achieving notable classification accuracy and perfect unsupervised clustering.**

***Index Terms*—Image Classification, Machine Learning, Transfer Learning, MobileNetV2, k-Nearest Neighbors, Agglomerative Clustering, Feature Extraction, Computer Vision**

I. Introduction

The relentless proliferation of digital imaging devices and online platforms has led to an exponential growth in image collections, presenting a significant challenge in terms of efficient management, retrieval, and categorization. Manual review and annotation of these vast datasets are not only exceedingly time-consuming but often practically infeasible, especially for large-scale digital archives, dynamically evolving online galleries, and extensive e-commerce visual assets. Project directly addresses this pressing issue by developing and proposing a comprehensive, end-to-end pipeline designed for the automatic categorization of images. This work aims to provide an accessible and effective solution for bringing order to visual data chaos.

A cornerstone of this project is the systematic demonstration and comparative analysis of both unsupervised and supervised machine learning paradigms applied to the image categorization task. Unsupervised learning, implemented in this system via Hierarchical Agglomerative Clustering (HAC), automatically groups images based on their inherent visual similarities without reliance on pre-existing labels. This approach is invaluable for discovering natural, latent structures within visual data and for organizing collections where labels are scarce or unavailable. Conversely, supervised learning, utilizing a k-Nearest Neighbors (k-NN) classifier trained on a dataset with predefined category labels (e.g., landscape, vehicle, animal), enables the precise assignment of new, unseen images to these known categories, providing a direct measure of classification performance.

The robust foundation for both these learning approaches is the effective extraction of discriminative visual features from images. This is achieved through the application of transfer learning, specifically employing the pre-trained MobileNetV2 convolutional neural network (CNN) [1]. This strategy allows the system to leverage the powerful representational capabilities of a model trained on millions of diverse images, obviating the need for designing and training a complex deep learning model from scratch, thereby saving considerable time and computational resources. This dual-path demonstration (unsupervised and supervised) not only provides critical insights into optimal data organization strategies and classification efficacy but also offers a versatile and adaptable framework. Such a framework is particularly beneficial in real-world scenarios where incoming images may lack labels, allowing unsupervised methods to provide an initial, automated organization, which can then be refined or utilized by supervised models. This comprehensive study aims to contribute a clear, practical demonstration of building and evaluating such a hybrid image classification system, highlighting the synergistic potential of combining diverse machine learning techniques for enhanced image understanding and management.

II. Methodology

The AutoImageClassifier system processes images through a configurable pipeline defined in `config.yaml`. Initial data preparation involves organizing images by category (e.g., ‘landscape/‘, ‘car/‘, ‘animal/‘) for automatic label detection, followed by an automated split into training, validation, and test sets. All images are then preprocessed by resizing to 224x224 pixels, normalizing pixel values to the 0-1 range, and converting to RGB format. Feature extraction leverages a pre-trained MobileNetV2 [1] as a frozen feature extractor, accessed via PyTorch/torchvision [3], to generate dense feature vectors. For unsupervised learning, Hierarchical Agglomerative Clustering (HAC) [4] from scikit-learn [2] groups these features without labels, using configurable affinity and linkage methods. Supervised classification employs a k-Nearest Neighbors (k-NN) algorithm [5] from scikit-learn [2], trained on labeled features (standardized via `StandardScaler`), with configurable ‘k‘ and distance metrics. Extracted features and model outputs are saved in formats like ‘.npz‘ and ‘.joblib‘.

III. Evaluation and Results

The performance of the AutoImageClassifier pipeline was thoroughly evaluated both quantitatively and qualitatively, with a distinct focus on the efficacy of its supervised classification component and the structural insights provided by unsupervised clustering. This section details the implementation, presents the performance metrics, and discusses the visual outcomes.

A. Implementation Details and Outputs

The core of the system was implemented in Python, leveraging libraries such as scikit-learn [2] for machine learning algorithms and PyTorch/torchvision [3] for deep learning model access. The Agglomerative Clustering algorithm was configured to produce three clusters, reflecting the three predefined image categories. Parameters like the linkage method (e.g., Ward, which minimizes the variance of merged clusters) and affinity metric (e.g., Euclidean distance) were specified in the central `config.yaml` file, allowing for easy experimentation. The primary outputs from this stage are cluster assignments for each image in the training and test sets, saved in NumPy's '.npz' format (e.g., `results/train_clusters.npz`).

For the supervised task, the k-NN classifier was "trained" by fitting it to the training feature vectors and their corresponding true category labels. This process for k-NN primarily involves storing the training data in an efficient structure for subsequent nearest neighbor searches. The number of neighbors ('k') and the distance metric were also configurable. The complete trained k-NN model, inclusive of the fitted `StandardScaler` used for feature normalization, is serialized and saved (e.g., as `results/classification_model.joblib` using joblib for efficient storage of scikit-learn objects). Predictions for the test set images are then generated using this trained model and stored (e.g., in `results/classification_results/test_preds`) for comprehensive performance assessment. This modular saving of intermediate and final results facilitates reproducibility and further analysis.

B. Classification Model Performance and Error Analysis

The k-NN model's generalization capability was rigorously assessed on the unseen test set. An overall accuracy of 0.78 was achieved, indicating a good general performance of the classifier. A detailed per-class analysis from the Classification Report provided deeper insights. The 'car' category was classified with perfect precision (1.00), recall (1.00), and F1-score (1.00), suggesting that the features extracted by MobileNetV2 are highly discriminative for vehicles and the k-NN model effectively learned this representation. The 'landscape' category also yielded strong results with a recall of 1.00 (all actual landscapes were identified) and an F1-score of 0.75, though its precision was 0.60, implying some non-landscape images were incorrectly classified as landscapes.

The 'animal' category presented the most challenge, exhibiting perfect precision (1.00) but a notably lower recall of 0.33 (F1-score 0.50). This specific outcome means that while every image the model predicted as 'animal' was indeed an animal, the model failed to identify two-thirds of the actual animal instances present in the test set. A closer inspection of misclassified 'animal' instances revealed that these often involved images where the animal was not the dominant subject, appearing smaller or partially obscured against a prominent landscape background, leading the classifier to prioritize landscape features. This lower recall for 'animal' could thus be attributed to factors such as greater intra-class visual diversity, the challenge of identifying non-dominant subjects, or the inherent limitations of k-NN in handling such complex visual cues with a limited number of neighbors. The macro average F1-score for the classifier was 0.75. The Confusion Matrix, presented in Fig. 1, graphically illustrates these per-class predictions, clearly showing the correct classifications along the diagonal and highlighting misclassifications, consistent with this observation.

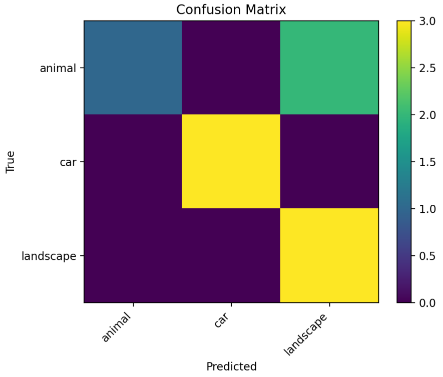

Fig. 1. Confusion matrix for the k-NN classifier on the test set. Rows represent actual classes, columns represent predicted classes.

C. Visualization and Unsupervised Clustering Performance

To provide an intuitive understanding of the system's behavior on individual instances, a Sample Image Grid (Fig. 2) was generated. This grid displays selected representative images from the test set, each annotated with its original visual form, the cluster ID assigned by the unsupervised Agglomerative Clustering algorithm, and the class label predicted by the supervised k-NN classifier. This side-by-side visualization allows for a rapid qualitative assessment of system behavior and facilitates direct comparison between the outcomes of the two learning paradigms.

A significant finding from the unsupervised learning phase was the exceptional performance of Agglomerative Clustering. When configured to produce three clusters, the algorithm achieved a perfect grouping of the test set images that directly and accurately corresponded to the three ground truth categories: animal, car, and landscape. This 100% correct clustering implies that the feature vectors extracted by MobileNetV2 for this specific dataset are highly separable in the feature space. Indeed, this high degree of separability is a direct consequence of leveraging a powerful, pre-trained model like MobileNetV2, which has learned rich and general visual representations from a vast dataset (ImageNet), making its features highly effective for downstream tasks such as this unsupervised grouping. The distinct nature of these categories, as captured by the deep features, allowed the unsupervised method to effectively discern the underlying class structure without any prior labeling information. This strong result from HAC underscores the quality of the feature representation and suggests that, for datasets with well-separated classes, unsupervised methods can be remarkably effective for initial data organization and discovery. The perfect clustering also provides a strong baseline and highlights that the k-NN classifier's errors are not due to fundamentally inseparable features but rather due to its own decision-making process or parameterization.

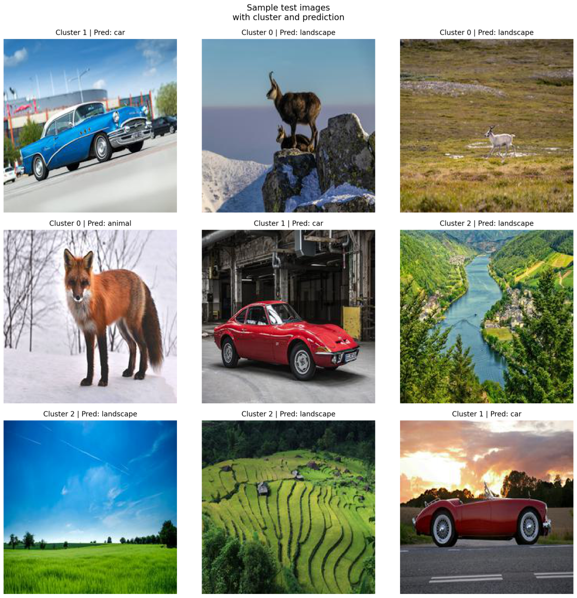
Sample test images
with cluster and prediction
Cluster 1 | Pred: car
Cluster 0 | Pred: landscape
Cluster 0 | Pred: landscape
Cluster 0 | Pred: animal
Cluster 1 | Pred: car
Cluster 2 | Pred: landscape
Cluster 2 | Pred: landscape
Cluster 2 | Pred: landscape
Cluster 1 | Pred: car

Fig. 2. Sample images from the test set showing the original image, assigned cluster ID (unsupervised), and predicted class (supervised k-NN).

D. Prediction Pipeline for New Images

The developed system culminates in a functional end-to-end pipeline capable of categorizing a novel, unseen image. This sequential process involves: 1) Standard preprocessing of the input image (resizing to 224x224, normalization of pixel values, and conversion to RGB format). 2) Extraction of its high-level feature vector using the established frozen MobileNetV2 setup. 3) Assigning the image to a cluster using the trained Agglomerative Clustering model to understand its natural grouping. 4) Finally, classifying the image into one of the predefined categories (landscape, vehicle, or animal) using the trained k-NN model. This demonstrates the practical applicability and utility of the implemented solution for real-world image categorization tasks.

IV. Conclusion

The ”AutoImageClassifier” project successfully designed, implemented, and evaluated an end-to-end pipeline for automatic image categorization, effectively demonstrating the capabilities of both unsupervised and supervised machine learning techniques. The strategic use of transfer learning with MobileNetV2 proved highly effective for robust feature extraction, significantly reducing development overhead while ensuring high-quality image representations suitable for subsequent learning tasks. The supervised k-NN classifier achieved a notable overall accuracy of 0.78 on the test set, exhibiting strong performance for 'car' and 'landscape' categories, while also identifying specific areas for improvement, particularly in the recall for the 'animal' class. Remarkably, the unsupervised agglomerative clustering perfectly aligned with the ground truth categories, indicating exceptionally separable features for this dataset and the potential of unsupervised methods for initial data structuring. This configurable and modular system provides a solid foundation for automated image organization and offers valuable comparative insights into the distinct strengths and applications of different machine learning paradigms in computer vision. Future work could fruitfully explore the expansion to a broader range of image categories, systematic hyperparameter optimization for both classifiers, and investigation into advanced feature extraction models or alternative learning algorithms.

References

[1] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted residuals and linear bottlenecks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 4510–4520.

[2] F. Pedregosa et al., “Scikit-learn: Machine learning in Python,” *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.

[3] A. Paszke et al., “PyTorch: An imperative style, high-performance deep learning library,” in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 32, 2019, pp. 8026–8037.

[4] F. Murtagh and P. Contreras, “Algorithms for hierarchical clustering: an overview,” *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 2, no. 1, pp. 86–97, Jan./Feb. 2012.

[5] N. S. Altman, “An introduction to kernel and nearest-neighbor nonparametric regression,” *The American Statistician*, vol. 46, no. 3, pp. 175–185, Aug. 1992.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2009, pp. 248–255.

[7] C. R. Harris et al., “Array programming with NumPy,” *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020.

[8] J. D. Hunter, “Matplotlib: A 2D graphics environment,” *Comput. Sci. Eng.*, vol. 9, no. 3, pp. 90–95, May/June 2007.

[9] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770–778.

[10] A. Dosovitskiy et al., “An image is worth 16x16 words: Transformers for image recognition at scale,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021.

[11] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[12] L. Van der Maaten and G. Hinton, “Visualizing data using t-SNE,” *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 25, 2012, pp. 1097–1105.