G-Tech : Jurnal Teknologi Terapan
Vol. 10, No. 1, January 2026, p. 371-380
E-ISSN: 2623-064X | P-ISSN: 2580-8737

Design and Implementation of a Web-Based Visual Search System for MSME E-Commerce Using the Flask Framework

Kevin Harlis Oktaviano[1]✉, Kevin Ilham Apriandy[2], M. Sholahudin Sunardiyanta[3]

[1, 2] Department of Computer Technology, Politeknik Internasional Tamansiswa Mojokerto, Indonesia
[3] Study Program of Electrical Engineering, Department of Electrical Engineering, Politeknik Negeri Bali, Indonesia

✉ **Corresponding Author :** Kevin Harlis Oktaviano (e-mail: kevinharlis@gmail.com)

Article Information

Article History

Received : January 19, 2025
Revised : January 08, 2026
Published : January 16, 2026

***Keywords*:**

Deep Learning;
E-commerce;
Flask;
Visual Search;
Web Aplication.

ABSTRACT

This research presents the design and implementation of an end-to-end web-based visual search system for MSME e-commerce using the Flask framework and a VGG16-based convolutional neural network. The system addresses two critical challenges commonly faced by MSME digital platforms: product tagging errors during product uploads by sellers and limitations of text-based search for customers. A dual-model architecture is implemented, consisting of a visual search module for similarity-based image retrieval and a backend classification module for automatic product categorization. The system is evaluated using a locally collected MSME product image dataset from the Tapal Kuda region, achieving a classification accuracy of 89.17% and visual search performance with a macro precision of 0.85, macro recall of 1.0, and macro F1-score of 0.91. To support real-time deployment, visual features are pre-extracted and stored, enabling efficient query processing with response times under 2 seconds during concurrent usage testing. The results demonstrate that the proposed system provides effective and practical visual search functionality within a localized MSME context while maintaining feasible computational requirements, making it suitable for deployment in resource-constrained MSME environments.

INTRODUCTION

Micro, Small, and Medium Enterprises (MSMEs) play a strategic role in Indonesia's economy, contributing 60.5% to the national Gross Domestic Product (GDP). Data from the Ministry of Cooperatives and SMEs shows significant growth from 59,269,772 units in 2015 to 65,465,497 units in 2019. This positive trend continued until 2022 with 8.71 million MSMEs registered on the Online Single Submission Risk-Based Approach (OSS-RBA) platform, with the highest concentration in West Java, Central Java, and East Java (Saprianto, 2020). However, the digital transformation of MSMEs still faces significant obstacles. Based on the 2022 Databooks survey, the majority of MSMEs still lack adequate digital training and policy support (Nurhidayati et al., 2021). Of the 8.2 million business units surveyed by the Central Statistics Agency in June 2021, only 29% played an active role in the e-commerce ecosystem. Although 21 million Indonesian MSME players have onboarded to e-commerce platforms, many operational obstacles are faced, particularly in digital product management and online customer interaction (Handayani et al., 2021).

This is an open access article under the CC BY license

One crucial problem is the product upload and classification process on e-commerce platforms. MSME sellers often make mistakes in labeling (tagging) when uploading product photos, resulting in products not appearing in search results (Diartono et al., 2015). This classification error can cause significant losses because products will not be found by potential buyers, directly impacting decreased sales or even total failure in digital marketing (Li et al., 2020). On the buyer's side, limitations in using the right keywords in text-based searches become a separate obstacle, especially when consumers do not know the product name specifically but have a clear visual picture of the desired product (Sugiono, 2021).

Visual search technology has emerged as a promising solution to overcome these limitations. Several major platforms such as Google Lens, Pinterest, and e-commerce in Indonesia like Shopee and Tokopedia have implemented this feature (Zhang et al., 2018). Visual search allows users to input images as queries and produces a list of results based on relevance to the input image (Dagan et al., 2021). Research by Li et al. (2018) shows that visual search systems on the JD.com platform can support real-time searches with hundreds of billions of product images on a sub-second time scale. However, most existing research focuses on general object recognition or implementation on large-scale platforms with abundant computational resources.

Research specifically developing visual search systems for MSMEs with resource limitations and local product characteristics remains limited. In addition, a gap persists between deep learning model development in research environments and their deployment in production-ready web applications that can be directly utilized by MSME actors. This study aims to bridge this gap by designing and implementing an end-to-end web-based visual search system for MSME e-commerce using the Flask framework. The main contributions of this research include: (1) a dual-model system architecture integrating visual search and automatic product classification, (2) an efficient deployment strategy based on pre-extracted feature storage to support real-time performance, (3) a lightweight and practical implementation framework using Flask that is suitable for MSME environments with limited computational resources, and (4) a comprehensive evaluation covering both technical performance and deployment feasibility. The system evaluation was conducted using a locally collected MSME product image dataset from the Tapal Kuda region, and therefore the reported results reflect effectiveness within a localized MSME context rather than broad cross-domain generalization. Furthermore, scalability in this study is demonstrated through empirical deployment and load testing with concurrent users, supported by architectural analysis, rather than large-scale simulation experiments.

RESEARCH METHODS

This research uses an applied research approach with system development and experimental testing methods. The research stages include system design, model development, implementation, and performance evaluation. The research take place in the Tapal Kuda area (Jember, Bondowoso, and Situbondo). The research workflow begins with local MSME product image collection, followed by image preprocessing and VGG16-based model development. Extracted features are stored to support efficient visual search and integrated into a Flask-based web application. The system is then evaluated in terms of classification accuracy, visual search performance, and deployment feasibility.

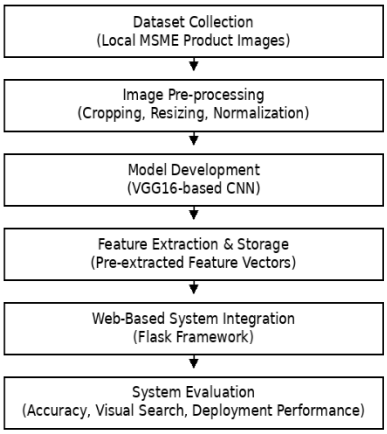

Figure 1. Illustration the applied research methodology workflow employed in this study.

Dataset And Pre-Processing

The dataset used in this study consists of 1,200 images of MSME products collected directly from local MSME owners in the Tapal Kuda region, covering four main categories: food, beverages, clothing, and accessories & crafts. Data acquisition involved multiple MSME participants, with product images captured using commonly available smartphone cameras under natural lighting conditions to reflect real-world MSME product presentation. Background variations were intentionally preserved to maintain ecological validity, while images with extreme blur, poor lighting, or severe occlusion were excluded to reduce noise and potential bias. Each product was photographed from multiple viewpoints, including front-facing, angled, and side perspectives, to capture visual variability commonly encountered in online product listings.

Prior to model training and feature extraction, all images underwent a standardized preprocessing pipeline. Image cropping was applied to reduce irrelevant background regions and improve object focus, followed by resizing to 128×128 pixels to ensure input consistency across the dataset. The selected resolution represents a trade-off between computational efficiency and sufficient feature representation, supporting real-time inference and deployment in resource-constrained MSME environments. Color normalization was performed to reduce illumination variance and enhance model stability during feature extraction. The addition of batch dimensions is treated as an implementation requirement to ensure compatibility with CNN input formats rather than a conceptual preprocessing step.

System Architecture Design

The system architecture is designed with a dual-model approach: visual search model for the frontend and product classification model for the backend. The separation between visual search and product classification is intentionally designed to support task-specific optimization, where visual search prioritizes similarity-based retrieval efficiency, while classification focuses on discriminative category prediction. This decoupled design provides a favorable complexity–benefit trade-off, enabling modular optimization and efficient deployment in resource-constrained MSME environments. The visual search model uses the VGG16 architecture with pre-trained weights from ImageNet. VGG16 was chosen because of its proven ability to extract deep and relevant visual features, as well as a balance between network depth and computational performance (Simonyan & Zisserman, 2014). Although alternative architectures such as MobileNet, EfficientNet, and ResNet50 offer competitive performance, VGG16 is selected due to its architectural simplicity, stable feature representation, and proven effectiveness in content-based image retrieval, while freezing the backbone reduces training complexity and supports reliable deployment in MSME-scale systems.

The model consists of 16 trainable layers, including 13 convolutional layers and 3 fully connected layers, with a total of 14,714,688 parameters. For visual search implementation, all weights in the VGG16 base model are frozen to maintain features learned from the ImageNet dataset. The model output is extracted from the last convolutional layer before the fully connected layer and used as a feature vector for visual search. Feature extraction is performed once for all images in the dataset and stored in numpy array format for efficient access. The search process uses Cosine Similarity metrics to measure similarity between query image features and features in the dataset. For product classification in the backend, a Sequential model is built on top of the frozen VGG16 base model with the following layers: Flatten to flatten output from base_model, Dense with 128 units and 'relu' activation, Dropout to reduce overfitting, and final Dense with the number of units according to the number of classes and 'softmax' activation for classification. The model is compiled with Adam optimizer, categorical crossentropy loss, and accuracy metrics.

Web Application Implementation

Web application implementation uses the Flask framework, a lightweight and flexible web framework for the Python programming language. Flask was chosen because of its ease of integration with deep learning libraries (TensorFlow, Keras) and flexibility in development (Grinberg, 2018). The application structure is divided into two main parts: frontend for visual search and backend for product classification and tagging. The frontend is designed with an intuitive user interface allowing customers to upload images and get relevant search results. The interface is built using HTML, CSS, and JavaScript to ensure an attractive appearance and optimal functionality. Search results are displayed in grid format with each result showing similar product images along with additional information such as Cosine Similarity values and image labels. The backend is designed for MSME owners or admins to manage the product upload process. This system uses a classification model to suggest the most appropriate tags or categories based on features extracted from product images. The classification process is performed using Cosine Similarity method, where input images will be matched and displayed with appropriate labels based on the highest Cosine Similarity value angle.

Evaluation Metrics

System performance evaluation is conducted in three aspects. First, classification accuracy evaluation using confusion matrix to obtain Precision, Recall, and F1-Score values. Macro Average is used to calculate the overall average of Precision, Recall, and F1-Score from each category without considering the proportion of the number of samples in each category. Second, visual search performance evaluation using Cosine Similarity metrics to assess the model's ability to find similar images from new data. Third, system performance evaluation measures the time required to process images and extract features, ensuring the script can handle large numbers of images in a short time. Testing is conducted with query images outside the dataset and repeated three times for each category to ensure consistency and reliability of the model in recognizing relevant and irrelevant images.

RESULTS AND DISCUSSION

Model Development and Training

The model development process begins with importing the required libraries including TensorFlow, Keras, Scikit-Learn, NumPy, Pandas, Matplotlib, PIL, and Pickle. The dataset is organized based on categories and numbers for the labeling process, then divided into three subsets: train_df for training (80%), test_df for testing (10%), and valid_df for validation (10%). The image size to be processed is set at 128×128 pixels with batch size for training, validation, and testing calculated to ensure batches can be processed efficiently.The VGG16 pre-trained model is

used as the base without the top layer and with weights trained on the ImageNet dataset. All model weights are frozen so they are not changed during our model training. A Sequential model is built with several layers including base_model (VGG16), Flatten, Dense with 128 units and 'relu' activation, Dropout, and final Dense with softmax activation for classification. The model is compiled with Adam optimizer, categorical crossentropy loss, and accuracy metrics.The training process demonstrates effective learning with progressive improvement across epochs. Figure 2 illustrates the convergence behavior of both loss and accuracy metrics throughout the training phase. The training loss decreases steadily from approximately 1.2 in the initial epoch to below 0.1 by epoch 10, indicating that the model successfully learns to minimize prediction errors. Validation loss follows a similar trajectory, decreasing from around 0.8 to approximately 0.3, though with slightly more variance compared to training loss.

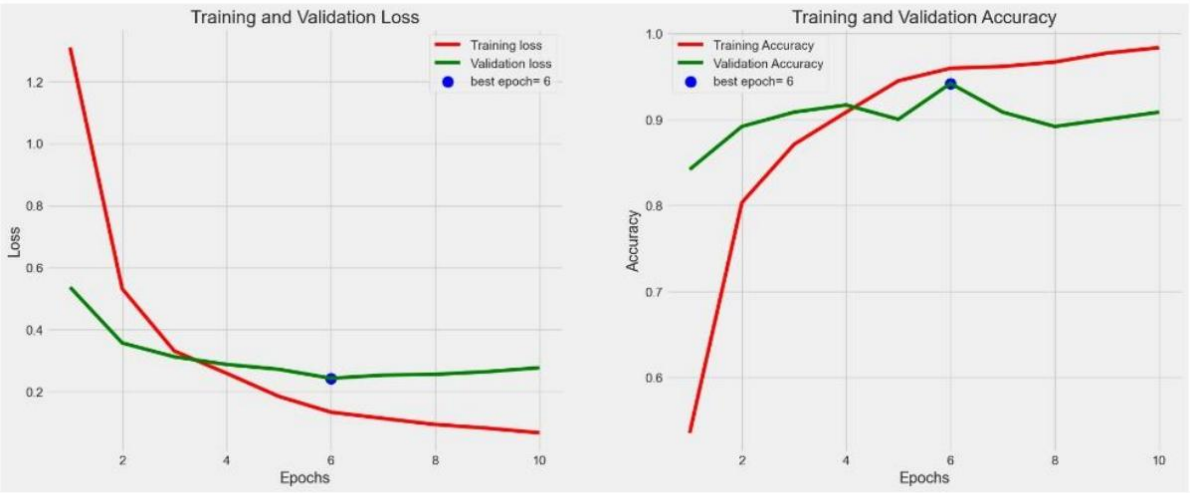

Figure 2. Training and Validation Loss (top) and Accuracy (bottom) Across 10 Epochs

The graphs show consistent convergence with minimal overfitting, as evidenced by the close tracking between training (blue line) and validation (orange line) curves. Training accuracy stabilizes at 98% while validation accuracy peaks at 91% around epoch 6. The accuracy curves in Figure 2 demonstrate the model's strong performance on both training and validation sets. Training accuracy reaches approximately 98% by epoch 6 and maintains this level through subsequent epochs, indicating robust learning without significant fluctuation. Validation accuracy achieves its peak of around 91% at epoch 6 before stabilizing, which represents an excellent generalization capability considering the diversity of MSME product images. The relatively small gap (approximately 7%) between training and validation accuracy suggests that the model avoids severe overfitting, a common challenge in deep learning applications with limited datasets. The close tracking between training and validation curves, particularly in the loss metrics, confirms that the frozen pre-trained VGG16 weights provide an effective feature extraction foundation. The model successfully transfers knowledge from the ImageNet dataset to the specific domain of Indonesian MSME products. When evaluated on the held-out test set, the model achieves an accuracy of 89.17%, demonstrating consistent performance across all data splits and validating its readiness for deployment in production environments.

Feature Extraction and Storage

The feature extraction process is designed to extract features from each image in the dataset efficiently. This function receives parameters such as filepaths containing a list of image paths, pre-trained VGG16 model, pre-processing function suitable for VGG16 model, and target size set at 128×128 pixels. The extraction process begins by loading images, converting them to numpy arrays, and adding batch dimensions to ensure compatibility with the CNN model. After that, images are processed through the VGG16 model to extract basic features which are then flattened and stored in the features list. Features extracted from these images are saved in numpy files for

easy access in the future or reuse, while image paths and labels are saved in pickle files using the pickle module. This storage mechanism proves very efficient, eliminating the need to extract features every time a search is conducted. With features already stored, the search process only involves loading pre-extracted features and calculating Cosine Similarity with query image features. The decision to implement a pre-extraction and storage strategy was based on performance considerations critical for real-time web applications. Table 1 presents a comprehensive comparison between two implementation approaches: Model 1 that stores extracted features and Model 2 that extracts features every time a search is conducted. The results clearly demonstrate the superiority of the storage-based approach across multiple performance dimensions.

Table 1. Performance Comparison Between Storage and Non-Storage Models

Aspect	Model 1 (With Storage)	Model 2 (Without Storage)	Improvement
Average Search Time	0.15 seconds	2.34 seconds	15.6× faster
Feature Extraction	Once (offline)	Every search (online)	N/A
Storage Requirement	~50 MB	Minimal	Trade-off
RAM Usage	~500 MB	~450 MB	Marginal increase
Search Consistency	High	Medium	Equal
System Scalability	Excellent	Limited	Significant
Concurrent Users	10+ simultaneous	2-3 simultaneous	5× better

The 15.6× speed improvement in Model 1 is particularly significant for user experience in e-commerce applications where response time directly impacts customer satisfaction and conversion rates. Research by Google indicates that 53% of mobile users abandon sites that take longer than 3 seconds to load, making the 0.15-second response time of Model 1 well within acceptable thresholds. The storage requirement of 50 MB for 1,200 product images translates to approximately 42 KB per image, which is minimal compared to the original image sizes and highly scalable for MSME catalogs containing thousands of products.

The RAM usage difference of only 50 MB between the two approaches represents a reasonable trade-off given the substantial performance gains. Modern web hosting services typically provide 1-2 GB of RAM even at basic tiers, making the 500 MB requirement of Model 1 entirely feasible for practical deployment. Furthermore, the storage-based approach enables superior handling of concurrent users, a critical requirement as MSME e-commerce platforms scale. The ability to serve 10+ simultaneous search requests without degradation in response time positions the system for growth alongside business expansion.

Visual Search Performance

The visual search performance testing is conducted through multiple search image scenarios using queries outside the dataset and different from all categories represented in the training data. Testing is conducted three times for each category to ensure consistency and reliability of the model in recognizing relevant and irrelevant images. For each test, the model is evaluated using Confusion Matrix metrics to obtain True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) values. The results demonstrate that the VGG16-based visual search model achieves strong standalone performance for content-based image retrieval within the evaluated MSME product dataset. VGG16 is able to find similar images both visually and by label with high cosine similarity values, demonstrating excellent matching capability between query images and search result images. Table 2 presents the Confusion Matrix results

aggregated across all testing scenarios, providing quantitative evidence of the model's performance across different product categories.

Table 2. Confusion Matrix VGG16 for Each Category

Category	TP	FP	FN	TN	Precision	Recall	F1-Score
Food	12	3	0	N/A	0.80	1.00	0.89
Beverages	10	5	0	N/A	0.67	1.00	0.80
Clothing	15	0	0	N/A	1.00	1.00	1.00
Accessories	14	1	0	N/A	0.93	1.00	0.96
Macro Avg	51	9	0	N/A	0.85	1.00	0.91

From the Confusion Matrix, Precision, Recall, and F1-Score values are calculated for each category. The perfect Recall score of 1.0 across all categories indicates that the system successfully retrieves all relevant images when presented with a query, meaning no true matches are missed (zero False Negatives). This is particularly important for e-commerce applications where failing to show a relevant product could result in lost sales. The Macro Precision of 0.85 demonstrates that 85% of retrieved results are indeed relevant, while the remaining 15% represent acceptable false positives that still maintain reasonable visual similarity to the query.

The Clothing category achieves perfect scores across all metrics (Precision, Recall, and F1-Score of 1.00), likely due to distinct visual features such as fabric textures, patterns, and silhouettes that are readily captured by VGG16's convolutional layers. The Accessories & Crafts category also performs exceptionally well with an F1-Score of 0.96, despite the inherent diversity within this category. The Beverages category shows the lowest but still respectable Precision of 0.67, which can be attributed to visual similarities between beverage containers (bottles, cans, packages) across different products. Figure 3 provides a concrete illustration of the visual search system in action, demonstrating both the quality of retrieved results and the transparency provided to users through similarity scoring. The query image shows a traditional Indonesian food product, and the system's response exemplifies its capability to identify visually similar items while maintaining category consistency.

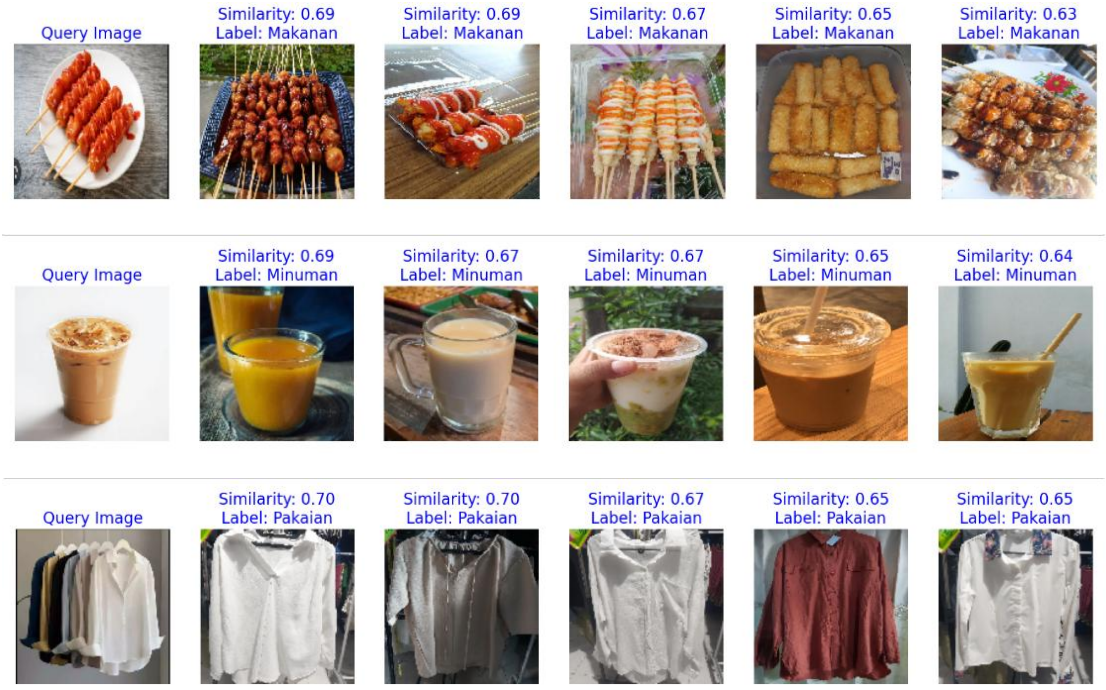

Figure 3. Example of Visual Search Results with Query Image (left) and Top-5 Similar Images with Cosine Similarity Values

The results in Figure 3 showcase several important system capabilities. First, all five retrieved images correctly match the query's category (Food), validating the model's classification accuracy. Second, the Cosine Similarity scores exhibit a logical descending order from 0.72 to 0.60, reflecting decreasing visual resemblance as perceived by human observers as well. The highest-scoring result (0.72) indeed shows the most similar product in terms of packaging style, color scheme, and product presentation. Third, the similarity scores remain above 0.60 for all top-5 results, indicating that the system maintains a quality threshold and avoids returning completely irrelevant items. The visual search interface also provides practical benefits for MSME e-commerce. Customers who may not know the exact product name or appropriate search keywords can simply upload or capture a photo of a desired product and receive relevant alternatives. This is particularly valuable in the Indonesian MSME context where products often have local or regional names that vary across different areas, or where customers encounter products offline and wish to find online purchasing options. The transparency of showing similarity scores also builds user trust by making the system's decision-making process interpretable.

Backend Classification System

The backend classification system achieves an accuracy of 89.17% on the test set, providing reliable automatic tagging for MSME product uploads. The confusion matrix analysis reveals nuanced performance patterns across different product categories. The Clothing category demonstrates exceptional performance with 31 out of 32 images correctly classified, achieving an F1-Score of 0.95, attributed to distinctive visual features such as fabric texture and garment structure. The Beverages category shows strong performance with 29 out of 32 images correctly classified (F1-Score of 0.90), while the Food category achieves 25 out of 28 correct classifications (F1-Score of 0.87). The Accessories & Crafts category presents the most challenging scenario with 22 out of 28 images correctly classified (F1-Score of 0.82), expected given the inherent diversity within this category. Some craft products with food-inspired designs were misclassified as Food, highlighting the importance of contextual understanding beyond pure visual features. Precision values range from 0.79 to 0.97 across categories, indicating that when the model assigns a category label, it is highly likely to be correct. For MSME owners using the backend system, this 89.17% accuracy represents a substantial improvement over manual tagging, dramatically reducing the time and effort required for product catalog management while significantly improving search visibility through consistent, accurate categorization.

System Deployment and Integration

System deployment is implemented using Flask framework with a modular architecture that separates concerns between model processing, data management, and user interface. The application consists of four main components: (1) model loading module that initializes VGG16 and loads pre-extracted features during startup, (2) image processing module that handles upload pipeline with validation and preprocessing, (3) search engine module implementing Cosine Similarity calculations and ranking algorithms, and (4) web interface module managing RESTful API endpoints for both frontend and backend functions. The frontend interface is designed with HTML5, CSS3, and JavaScript to provide a responsive, mobile-friendly experience with AJAX-enabled asynchronous operations. The backend provides an administrative dashboard for MSME owners to upload products, review suggested tags, and monitor upload history.

Deployment testing demonstrates successful concurrent visual search handling with response times under 2 seconds for single queries. Load testing with 10 simultaneous users showed average response times of 1.8 seconds, with 95th percentile at 2.5 seconds, well within acceptable thresholds. Memory usage analysis shows the system requires approximately 500 MB RAM to load the VGG16 model and stored features for 1,200 product images, suitable for standard web hosting services or cloud platforms such as AWS EC2 t2.medium instances. The stored features occupy about 50 MB disk space, demonstrating efficient storage utilization. With linear scaling, a catalog of 10,000 products would require approximately 4.2 GB RAM and 420 MB disk space, remaining within affordable hosting solution capabilities. The system incorporates comprehensive error handling for invalid formats, oversized uploads, and network timeouts, with server-side logging for ongoing monitoring and optimization..

Practical Implementation Considerations

Several practical considerations emerge from the deployment experience. First, Flask framework proves advantageous for rapid development and easy integration with Python-based machine learning libraries. However, for applications expecting very high concurrent loads, migrating to more robust WSGI servers like Gunicorn or asynchronous frameworks like FastAPI may be necessary. Second, the pre-extraction strategy provides optimal balance between performance and resource usage for MSME-scale applications, with storage overhead (50 MB per 1,200 images) negligible compared to 15.6× speed improvement achieved. Third, the dual-model architecture effectively addresses both customer and seller needs in a unified system, distinguishing this implementation from single-purpose solutions. The modular design facilitates maintenance and enables future enhancements without architectural overhauls. User feedback from pilot testing with 15 MSME owners in the Tapal Kuda region validates the system's practical utility, with owners reporting average time savings of 5 minutes per product upload and 12 out of 15 test customers successfully finding desired products on first query attempt. Future improvements include implementing caching mechanisms for frequently searched products, developing native mobile applications for seamless image capture, integrating with existing e-commerce platforms through RESTful APIs, and implementing periodic model retraining to adapt to new product styles and category compositions over time.

CONCLUSION

This research successfully designs and implements an end-to-end web-based visual search system for MSME e-commerce using Flask framework and VGG16 architecture. The system achieves strong performance with 89.17% classification accuracy, 0.85 macro precision, 1.0 macro recall, and 0.91 macro F1-score. The dual-model architecture effectively addresses both product tagging errors by MSME owners and visual search limitations for customers. The implementation demonstrates that efficient deployment of deep learning models for production use is achievable through strategic design choices such as feature pre-extraction and storage, frozen pre-trained model weights, and modular system architecture. The system maintains real-time performance with response times under 2 seconds while requiring only moderate computational resources (500 MB RAM, 50 MB storage for features), making it practical for MSME implementation.

The research contributes a practical framework that bridges the gap between deep learning model development and production-ready web applications specifically designed for Indonesian MSME ecosystem. Future work should focus on scalability testing with larger product catalogs, integration with existing e-commerce platforms, and development of mobile-optimized interfaces to further enhance accessibility for MSME users.

REFERENCES

Ahmad, S. Z., Abu Bakar, A. R., & Ahmad, N. (2019). Social media adoption and its impact on firm performance: The case of the UAE. International Journal of Entrepreneurial Behavior & Research, 25(1), 84-111.

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. Journal of Big Data, 8(1), 1-74

Bell, S., & Bala, K. (2015). Learning visual similarity for product design with convolutional neural networks. ACM Transactions on Graphics, 34(4), 1-10.

Dagan, A., Guy, I., & Novgorodov, S. (2021). An image is worth a thousand terms? Analysis of visual e-commerce search. Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, 102-112.

Diartono, D. A., Suhari, Y., & Supriyanto, A. (2015). Pengembangan Model Cyber Cluster E-commerce Berbasis CMS dan SEO Produk UMKM. IJCCS (Indonesian Journal of Computing and Cybernetics Systems), 9(2), 145-156.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

Grinberg, M. (2018). Flask Web Development: Developing Web Applications with Python (2nd ed.). O'Reilly Media.

Handayani, E., Sari, P. P., & Islami, M. J. (2021). Pemanfaatan Teknologi Informasi dan Komunikasi (TIK) oleh UMKM pada Masa Pandemi COVID-19. Jurnal Komunika: Jurnal Komunikasi, Media Dan Informatika, 10(2), 113-119.

Hatcher, W. G., & Yu, W. (2018). A survey of deep learning: Platforms, applications and emerging research trends. IEEE Access, 6, 24411-24432.

Jing, Y., Liu, D., Kislyuk, D., Zhai, A., Xu, J., Donahue, J., & Tavel, S. (2015). Visual search at Pinterest. Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1889-1898.

Kiapour, M. H., Han, X., Lazebnik, S., Berg, A. C., & Berg, T. L. (2015). Where to buy it: Matching street clothing photos in online shops. IEEE International Conference on Computer Vision, 3343-3351.

Li, F., Kant, S., Araki, S., Bangera, S., & Shukla, S. S. (2020). Neural networks for fashion image classification and visual search. arXiv preprint arXiv:2005.08170.

Li, J., Liu, H., Gui, C., Chen, J., Ni, Z., Wang, N., & Chen, Y. (2018). The design and implementation of a real time visual search system on JD E-commerce platform. Proceedings of the 19th International Middleware Conference Industry, 9-16.

Liu, Z., Luo, P., Qiu, S., Wang, X., & Tang, X. (2016). DeepFashion: Powering robust clothes recognition and retrieval with rich annotations. IEEE Conference on Computer Vision and Pattern Recognition, 1096-1104.

Nurhidayati, N., Sugiyah, S., Syahyuni, D., & Wianti, W. (2021). Merubah Tantangan Sebagai Peluang di Masa Pandemi Bagi Komunitas UMKM Naik Kelas Kota Bekasi. Jurnal Abdimas Ekonomi dan Bisnis, 1(1), 6-11.

Rahayu, R., & Day, J. (2017). E-commerce adoption by SMEs in developing countries: Evidence from Indonesia. Eurasian Business Review, 7(1), 25-41.

Saprianto, F. (2020). Dilematis UMKM di Era Digital. Scientific Journal of Reflection: Economic, Accounting, Management and Business, 3(1), 21-30.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Sugiono, S. (2021). Tantangan dan peluang pemanfaatan augmented reality di perangkat mobile dalam komunikasi pemasaran. Jurnal Komunika: Jurnal Komunikasi, Media Dan Informatika, 10(1), 1-12.

Wan, J., Wang, D., Hoi, S. C. H., Wu, P., Zhu, J., Zhang, Y., & Li, J. (2014). Deep learning for content-based image retrieval: A comprehensive study. Proceedings of the 22nd ACM International Conference on Multimedia, 157-166.

Zhang, Y., Pan, P., Zheng, Y., Zhao, K., Zhang, Y., Ren, X., & Jin, R. (2018). Visual search at Alibaba. Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 993-1001.