Contents lists available at Indonesian Scholar Society
Indonesian Journal of Data Risk Research
Journal Homepage:
https://journal.solusiriset.com/index.php/ijdrr/index
e-ISSN: xxxx-xxxx

Evaluation of the CLIP Architecture for Zero-Shot Image Classification on the Intel Image Classification Dataset

Ade Lailani*, Rohmi Dyah Astuti, Christyan Tamaro Nadeak

Data Science Study Program, Sumatra Institute of Technology, Indonesia

Corresponding Email: ade.lailani@sd.itera.ac.id

ABSTRACT

The performance evaluation of various architectures in the Contrastive Language–Image Pre-training (CLIP) model was conducted in a zero-shot image classification scenario. Image classification was performed using the Intel Image Classification Dataset, which consists of 3000 images representing several environmental categories. This study compares several CLIP architectures based on ResNet and Vision Transformer. Model performance was evaluated using accuracy, F1-score, precision, and recall metrics. The experimental results show that the RN50x16 architecture achieved the best performance with an accuracy of 0.925, an F1-score of 0.925, a precision of 0.929, and a recall of 0.925. The RN101, RN50x64, and ViT-B/32 architectures also demonstrated relatively strong performance with accuracy values around 0.92. In contrast, the ViT-B/16, ViT-L/14, and ViT-L/14@336px architectures produced lower performance with accuracy values below 0.90. Furthermore, the Mean Cosine Similarity Matrix analysis indicates that models with ResNet-based architectures produce clearer class representation separation compared to several Vision Transformer variants. Overall, the results suggest that the choice of architecture significantly influences the performance of the CLIP model in zero-shot image classification, with RN50x16 emerging as the most optimal architecture for the dataset used.

ARTICLE INFO

Received
11 Jan 2026
Revised
17 March 2026
Accepted
25 Mei 2026

Keywords: CLIP, Image Classification, ResNet, Vision Transformer, Zero-Shot.

I. INTRODUCTION

Computer vision is a field of image processing with one of its fundamental tasks being image classification, which aims to assign category labels automatically to an image. The rapid

development of deep learning in the last decade has significantly boosted the performance of image classification systems, including the use of convolutional architectures such as AlexNet [1], VGG19 [2], and ResNet [3]. The implementation of these architectures for image classification purposes has been widely applied in various domains, including remote sensing, autonomous navigation systems, and medical image analysis [4].

A new paradigm in image classification that is currently being developed is the vision-language-based model, such as CLIP (Contrastive Language–Image Pre-training), with a zero-shot learning approach. CLIP is trained using a massive amount of data from the internet in the form of text-image pairs, enabling it to classify images based on text descriptions without requiring additional training. This capability makes CLIP very flexible and efficient in image classification without the high computational costs [5].

The Intel Image Classification dataset [6] is a widely used benchmark dataset for classifying natural scenes, encompassing six categories: buildings, forests, glaciers, mountains, seas, and roads. This dataset is representative because it has a high visual variation between classes, as well as some classes that are visually similar, making it an interesting challenge to evaluate using various classification approaches [7].

Several previous studies have evaluated the performance of CNN models on this dataset. Research by [8] shows that ResNet50 is capable of achieving an accuracy above 90% on the Intel dataset using transfer learning techniques. Meanwhile, a comparative study by [9] compared several CNN architectures and found that ResNet provided the best results compared to VGG and MobileNet. However, there has not been much research using a zero-shot approach based on vision-language models like CLIP on the same dataset [10].

The architecture of the CLIP model essentially consists of two encoders, namely the image encoder and the text encoder. The difference lies in the architecture of the image encoder used, which can be either ResNet or Vision Transformer [11], [12], [13], [14]. The main differences between the variants of Vision Transformer lie in the model size (Base or Large), patch size, and input image resolution, where smaller patches and higher resolutions generally produce more detailed visual representations and improve zero-shot classification performance [13], [15].

The purpose of this study is to evaluate the performance of various architectures on the CLIP model in performing zero-shot-based image classification on the Intel Image Classification Dataset, as well as to analyze the performance differences between the ResNet and Vision Transformer architectures to determine the most effective model in understanding visual representations without additional training processes.

II. METHODS

1. ResNet50 and Transfer Learning

ResNet (Residual Network) was introduced by [3] as a solution to the vanishing gradient problem in very deep neural networks. Its main innovation is the residual connection (skip connection) that allows the gradient to flow directly through several layers. ResNet50 is a variant with 50 layers that has proven effective in various image classification tasks [3].

Transfer learning is a technique that uses the weights of a model trained on a large dataset, such as ImageNet, to adapt to a new task [16]. By freezing the initial layers that capture general features and only retraining the final layers (fine-tuning), the model can achieve high performance even with relatively limited training data [4].

2. CLIP (Contrastive Language–Image Pre-training)

CLIP was developed by OpenAI, as described in [15], by training image and text encoders simultaneously using a contrastive loss on 400 million text-image pairs from the internet. This architecture allows the model to learn joint representations between visual and textual modalities.

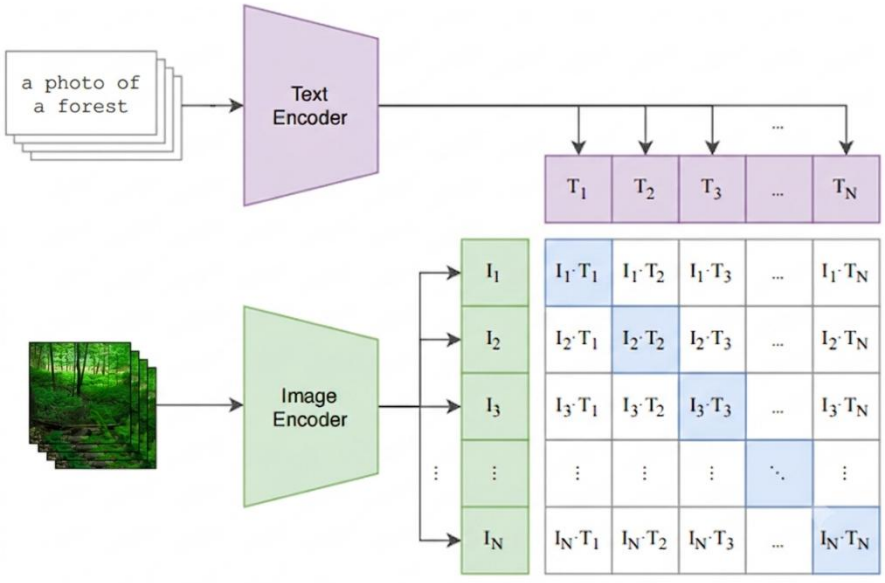

Figure 1. CLIP Model with Zero-Shot [15]

In zero-shot inference in Figure 1. CLIP calculates the cosine similarity between image embeddings and text embeddings from each class prompt, then selects the class with the highest similarity score [15]. This approach has proven to be competitive with supervised models on various standard benchmarks [5].

In the CLIP architecture, each pair of images and text is processed separately through the image encoder and text encoder to generate feature representations from each modality. The

representations are then mapped into the same multimodal embedding space through a trained weight matrix and normalized using L2 normalization so that they can be directly compared with cosine similarity. After that, the similarity level between all pairs of images and texts in one batch is calculated, and then the results are scaled using the temperature parameter that was also optimized during the training process. Because each image is assumed to have one corresponding text at the same index position, the correct pair labels are created as a supervision reference [5].

The model then optimizes the contrastive loss bidirectionally, namely, how well the image can identify the corresponding text and vice versa. The final loss value is obtained from the average of both directions, so the model gradually learns to bring closer the representations of relevant pairs while distancing the representations of non-matching pairs in the shared embedding space [5].

CLIP Pseudocode

```
# image_encoder     - ResNet or Vision Transformer
# text_encoder      - CBOW or Text Transformer
# I[n, h, w, c]     - minibatch of aligned images
# T[n, l]           - minibatch of aligned texts
# W_i[d_i, d_e]     - learned proj of image to embed
# W_t[d_t, d_e]     - learned proj of text to embed
# t                 - learned temperature parameter

# extract feature representations of each modality
I_f = image_encoder(I) #[n, d_i]
T_f = text_encoder(T) #[n, d_t]

# joint multimodal embedding [n, d_e]
I_e = l2_normalize(np.dot(I_f, W_i), axis=1)
T_e = l2_normalize(np.dot(T_f, W_t), axis=1)

# scaled pairwise cosine similarities [n, n]
logits = np.dot(I_e, T_e.T) * np.exp(t)

# symmetric loss function
labels = np.arange(n)
loss_i = cross_entropy_loss(logits, labels, axis=0)
loss_t = cross_entropy_loss(logits, labels, axis=1)
loss = (loss_i + loss_t)/2
```

Figure 2. Numpy-Like Pseudocode for the Core of an Implementation of CLIP

3. Intel Image Classification Dataset

The Intel Image Classification dataset [6] consists of approximately 25,000 images sized 150×150 pixels, divided into six categories of scenery: buildings, forest, glacier, mountain, sea, and street. This dataset has been widely used as a benchmark in landscape image classification research [7].

4. Evaluation Metrics

The evaluation of classification model performance generally uses accuracy, precision, recall, and F1-score, as well as the Mean Cosine Similarity Matrix [17]. The weighted F1-score and the Mean Cosine Similarity Matrix are chosen as the main metrics because they are more informative for imbalanced class distributions. The confusion matrix is used to analyze the patterns of classification errors between classes in more detail [18].

5. Dataset and Data Pre-processing

The dataset used is the Intel Image Classification [6] downloaded from Kaggle using kagglehub. The dataset used is the test set (3,000 images) in six classes. In the variation of the CLIP architecture, each preprocessing step uses the built-in functions of the ResNet and Vision Transformer architectures, which include resizing, center cropping, and CLIP-specific normalization.

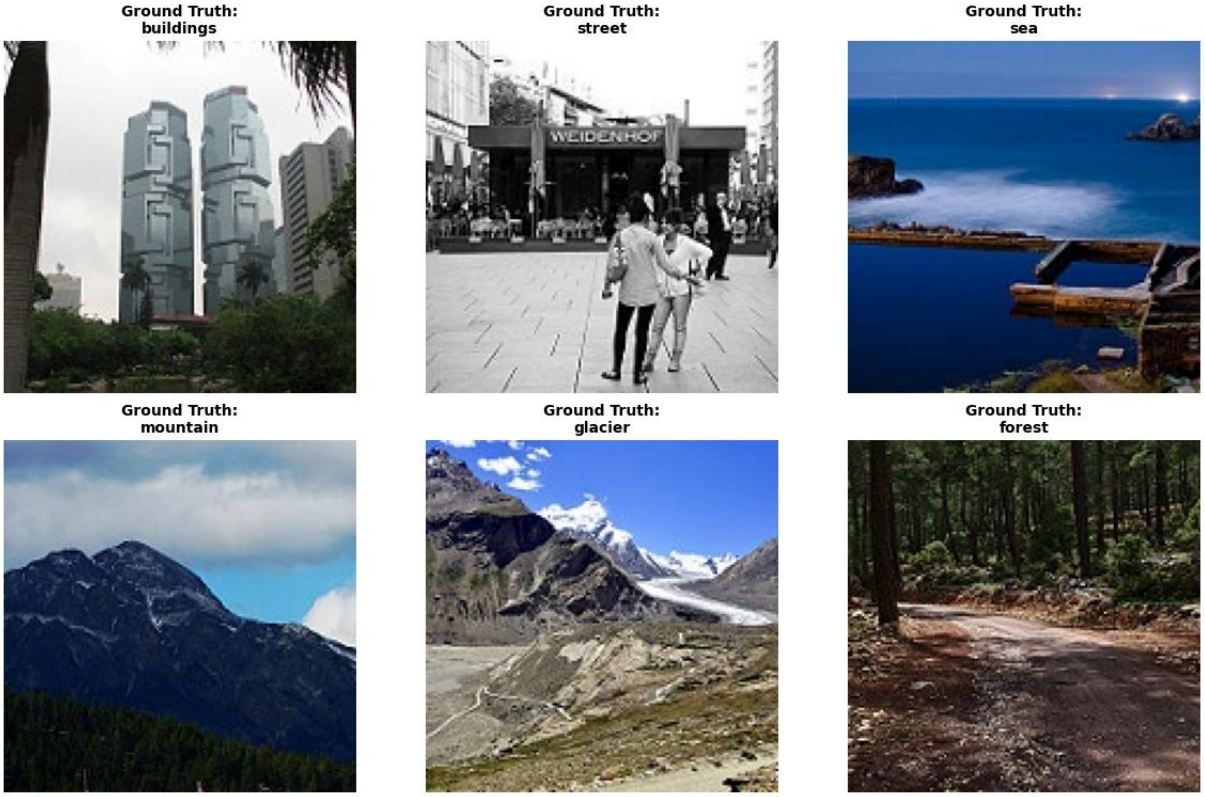

Figure 3. Raw Dataset Samples Before Entering CLIP

6. Model Architecture and Configuration

a. CLIP Zero-Shot

The CLIP ViT-B/32 model is used without weight modifications. Classification is performed by encoding a simple text prompt for each class (e.g., "A photo of buildings") using the CLIP text

encoder, then calculating the cosine similarity between the image embedding and the text embedding of the six classes. The class with the highest cosine similarity score is selected as the final prediction. There are 6 classes in the dataset used, and the text prompt is constructed according to the class names listed in Table 1.

Table 1. Text Prompts Used in Zero-Shot CLIPs

Class	Text Prompt
Buildings	A photo of buildings
Forest	A photo of a forest
Glacier	A photo of a glacier
Mountain	A photo of a mountain
Sea	A photo of the sea
Street	A photo of a street

b. CLIP Architecture Variations

The architecture of the CLIP model essentially consists of two encoders: an image encoder and a text encoder. The difference lies in the architecture of the image encoder used, which can be either ResNet or Vision Transformer. Here is a brief explanation of each architecture written in Table 2.

Table 2. CLIP Zero-Shot Architecture Variation

Architecture	Characteristics
RN101	CNN with 101 layers
RN50x4	RN50 magnified 4×
RN50x16	RN50 magnified 16×
RN50x64	RN50 magnified 64×
ViT-B/32	Base model, patch 32
ViT-B/16	Base model, patch 16
ViT-L/14	Large model, patch 14
ViT-L/14@336px	Large model, input resolution 336

c. Data Analysis Techniques

The evaluation of both models was conducted on the same test set using the following metrics: (1) Overall accuracy; (2) Weighted F1-score; (3) Precision and recall per class; and (4) Normalized confusion matrix. All experiments were conducted on devices with GPUs using the PyTorch 2.0 framework and the OpenAI CLIP library.

This study uses an experimental design comparing several CLIP architectures based on ResNet and Vision Transformer with a zero-shot approach. The dependent variables measured are

accuracy, weighted F1-score, and the distribution of predictions per class, visualized through a confusion matrix.

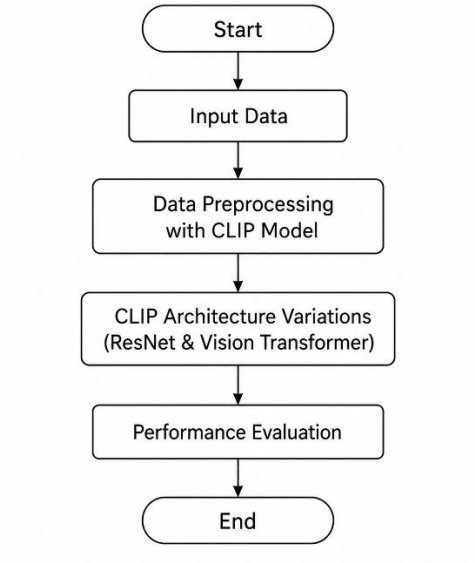

Figure 4. Research Flowchart

Variations in model architecture are evaluated on the same test set to ensure a fair and consistent comparison. The research flowchart in Figure 2 starts from data input and preprocessing, followed by variations of the CLIP architecture, and then a comparison of model performance evaluation. In the CLIP model, several image encoder architectures are used to extract visual representations from images. The RN101, RN50x4, RN50x16, and RN50x64 variants are developments of the ResNet architecture, where capacity is increased by enlarging the depth or width of the network, enabling it to capture more complex visual features. Meanwhile, ViT-B/32, ViT-B/16, ViT-L/14, and ViT-L/14@336px use the Vision Transformer architecture, which divides images into patches and processes them using a self-attention mechanism to capture global relationships between parts of the image.

III. RESULTS AND DISCUSSION

1. CLIP Architecture Performance Evaluation

The evaluation results of the CLIP model's performance in the zero-shot image classification scenario are shown in Table 3. The evaluation was conducted to compare several CLIP architectures using accuracy and F1-score (weighted) metrics. Both metrics are used to measure the classification accuracy and the model's performance balance in recognizing various classes in the dataset.

Tabel 3. Accuracy and F1-score Evaluation Results of CLIP Zero-Shot architecture

Architecture	Accuracy	F1-Score (w)

RN101	0.922	0.922
RN50x4	0.899	0.897
RN50x16	0.925	0.925
RN50x64	0.919	0.919
ViT-B/32	0.920	0.919
ViT-B/16	0.889	0.888
ViT-L/14	0.892	0.889
ViT-L/14@336px	0.891	0.887

Based on the experimental results on the CLIP model using the Intel Image Classification Dataset, it is evident that the zero-shot image classification performance varies across different architectures. The RN50x16 architecture produced the best performance with an accuracy of 0.925 and an F1-score of 0.925, indicating that the increased capacity in the ResNet family can provide better visual representations. The RN101 and ViT-B/32 architectures also showed quite high performance with accuracies above 0.92, making them fairly effective in understanding the relationship between images and text in a zero-shot scheme.

The ViT-B/16, ViT-L/14, and ViT-L/14@336px architectures exhibit lower performance, with accuracy below 0.90. This indicates that, in this dataset, the ResNet architecture in the CLIP model tends to be more stable than several variants of the Vision Transformer in a zero-shot scenario. Additionally, the nearly identical accuracy and F1-score values indicate a fairly balanced prediction distribution, with RN50x16 being the best architecture in this experiment.

Tabel 4. Precision and Recall Evaluation Results of the CLIP Zero-Shot Architecture

Architecture	Precision	Recall
RN101	0.922	0.922
RN50x4	0.9	0.899
RN50x16	0.929	0.925
RN50x64	0.920	0.919
ViT-B/32	0.922	0.920
ViT-B/16	0.899	0.889
ViT-L/14	0.898	0.892
ViT-L/14@336px	0.897	0.891

Table 4 displays the evaluation results on the CLIP model with variations of 8 architectures, where the precision and recall values show a performance pattern similar to the previous accuracy metrics. The RN50x16 architecture provides the best performance with a precision of 0.929 and a recall of 0.925, indicating the model's ability to make accurate predictions while also detecting most classes well. The RN101, RN50x64, and ViT-B/32 architectures also show quite high performance with precision and recall values around 0.92 and thus can be categorized as having stable classification capabilities.

The ViT-B/16, ViT-L/14, and ViT-L/14@336px architectures produced lower precision and recall values compared to other models. This indicates that the ResNet architecture in the CLIP model tends to provide better performance in zero-shot image classification, with RN50x16 being the most optimal architecture in this experiment.

2. Mean Cosine Similarity Matrix Evaluation

The evaluation of the Mean Cosine Similarity Matrix is conducted between the image class and the prompt class in the CLIP model using the Intel Image Classification Dataset. The values on the diagonal represent the similarity between the image and the correct class prompt, while the values off the diagonal indicate the potential for class confusion.

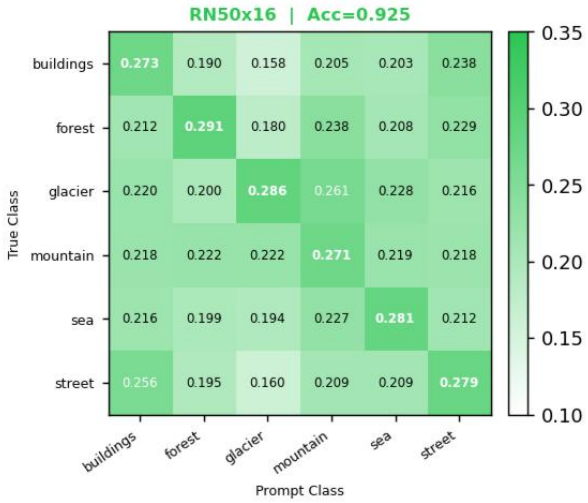

Figure 5. Mean Cosine Similarity Matrix of RN50x16 Architecture

In Figure 5, the RN50x16 architecture shows the best performance with an accuracy of 0.925, as seen from the relatively consistent and quite high diagonal values in most classes. This indicates that the model is capable of effectively mapping the visual representation of images to the corresponding text descriptions. The RN101 architecture (Figure 6), RN50x64 (Figure 7), and ViT-B/32 (Figure 8) also show a fairly clear diagonal pattern, resulting in an accuracy above 0.91.

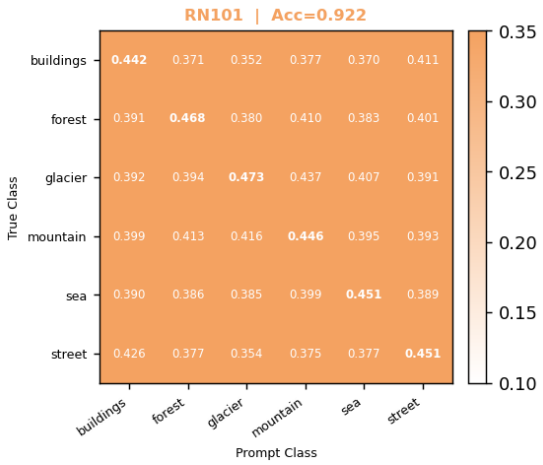

Figure 6. Mean Cosine Similarity Matrix of RN101 Architecture

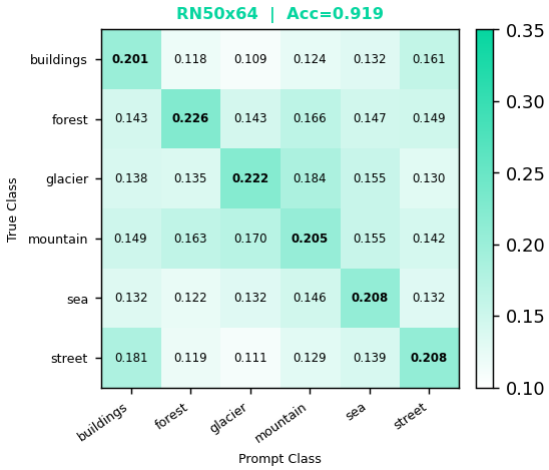

Figure 7. Mean Cosine Similarity Matrix of RN50x64 Architecture

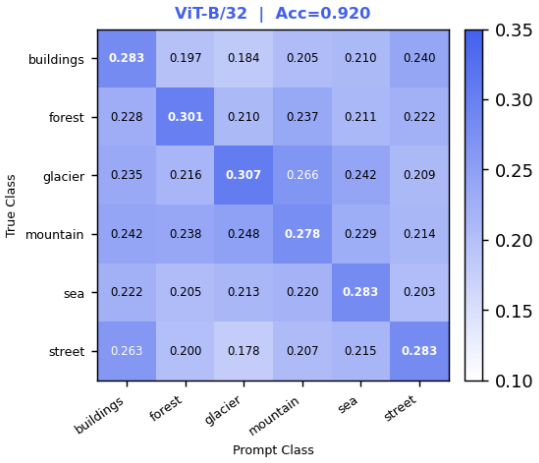

Figure 8. Mean Cosine Similarity Matrix of ViT-B/32 Architecture

On the other hand, the ViT-B/16 (Figure 9), ViT-L/14 (Figure 10), ViT-L/14@336px (Figure 11), and RN50x4 (Figure 12) architectures have lower diagonal values and more uniform off-diagonal values, indicating a higher level of class confusion. Some classes, such as mountain, glacier, and buildings, also show a fairly high similarity with other classes, potentially leading to classification errors.

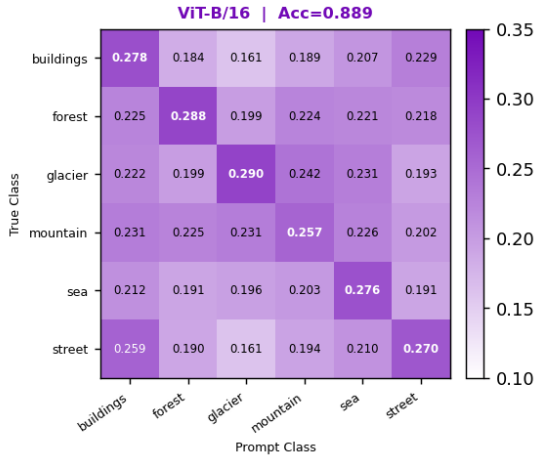

Figure 9. Mean Cosine Similarity Matrix of ViT-B/16 Architecture

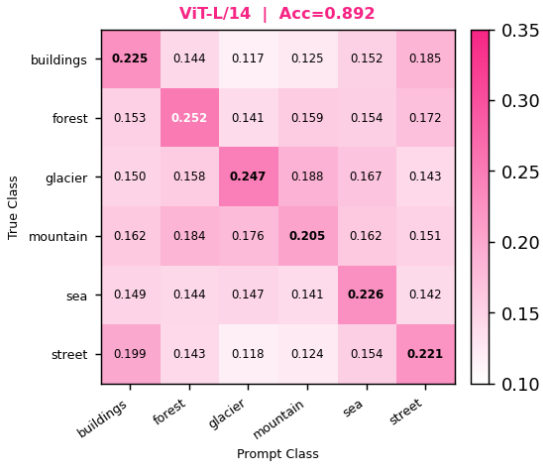

Figure 10. Mean Cosine Similarity Matrix of ViT-L/14 Architecture

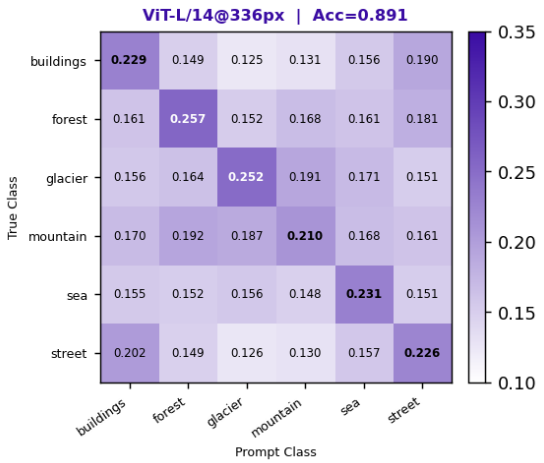

Figure 11. Mean Cosine Similarity Matrix of ViT-L/14@336px Architecture

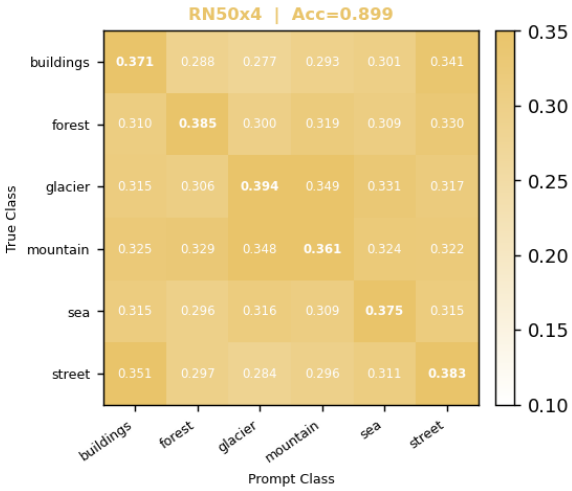

Figure 12. Mean Cosine Similarity Matrix of RN50x4 Architecture

Overall, this visualization shows that the ResNet-based architecture on CLIP tends to produce clearer class representation separations compared to several variants of Vision Transformer, which is consistent with the previous accuracy and F1-score evaluation results.

IV. CONCLUSION

Implementation of several architectures on the CLIP model for zero-shot image classification using the Intel Image Classification Dataset. The experimental results show that the RN50x16 architecture yields the best performance with the highest accuracy and F1-score compared to other architectures. The ResNet-based architectures in CLIP (RN101, RN50x4, RN50x16, and RN50x64) demonstrate more stable performance compared to several variants of Vision Transformer (ViT-B/16, ViT-L/14, and ViT-L/14@336px) in the zero-shot image classification scenario on the used dataset. The advantage of this approach is the ability of the CLIP model to

perform classification without retraining (zero-shot), making it more efficient in the use of labeled data. However, the model's performance is still influenced by the choice of architecture and the limitations of class representation in the text prompts used.

This study's limitations lie in the use of a simple prompt and a single dataset, so the evaluation results do not fully reflect CLIP's performance across a wider range of data domains. Future research is expected to expand the dataset types and combine other methods.

V. AUTHOR CONTRIBUTION

The authors contributed equally to the formulation of the research concept, data collection, data analysis, manuscript writing, and approval of the final manuscript for publication.

VI. CONFLIC OF INTEREST

The authors declare that there are no potential conflicts of interest related to the research, writing, or publication of this article.

ACKNOWLEDGMENT

With deep gratitude and appreciation, we would like to extend our heartfelt thanks to the various parties who have contributed to the research and writing of this article. This research would not have been possible without the help and support of many parties. We would like to express our gratitude to the Institute of Technology Sumatra and the Data Science Study Program for their adequate research facility support.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Commun. ACM*, vol. 60, no. 6, 2017, doi: 10.1145/3065386.

[2] M. R. Wedatama, L. J. E. Dewi, and N. W. Marti, “Klasifikasi pose yoga surya namaskar menggunakan algoritma convolutional neural network dengan arsitektur VGG19 dan ResNet-50,” *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 14, no. 1, 2026, doi: 10.23960/jitet.v14i1.8824.

[3] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016. doi: 10.1109/CVPR.2016.90.

[4] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, “A survey on deep transfer learning,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018. doi: 10.1007/978-3-030-01424-7_27.

[5] D. D. Putri, G. F. Nama, and W. E. Sulistiono, "Analisis sentimen kinerja dewan perwakilan rakyat (DPR) pada twitter menggunakan metode naive bayes classifier," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 10, no. 1, 2022, doi: 10.23960/jitet.v10i1.2262.

[6] P. Bansal, "Intel Image Classification." [Online]. Available: https://www.kaggle.com/datasets/puneet6060/intel-image-classification

[7] S. Bianco, R. Cadene, L. Celona, and P. Napoletano, "Benchmark analysis of representative deep neural network architectures," *IEEE Access*, vol. 6, 2018, doi: 10.1109/ACCESS.2018.2877890.

[8] R. Sinaga, A. Purnama, and B. Nugroho, "Klasifikasi citra pemandangan menggunakan ResNet50 dengan transfer learning," *J. Teknol. Inf.*, vol. 15, no. 2, pp. 45–53, 2022.

[9] D. Prasetyo, M. Fauzi, and S. Wahyuni, "Perbandingan arsitektur CNN untuk klasifikasi citra intel image classification," *J. Ilmu Komput. dan Inf.*, vol. 8, no. 1, pp. 12–21, 2023.

[10] M. Minderer *et al.*, "Revisiting the calibration of modern neural networks," in *Advances in Neural Information Processing Systems*, 2021.

[11] X. Zhai, A. Kolesnikov, N. Houlsby, and L. Beyer, "Scaling vision transformers," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2022. doi: 10.1109/CVPR52688.2022.01179.

[12] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A survey of transformers," *AI Open*, vol. 3, 2022, doi: 10.1016/j.aiopen.2022.10.001.

[13] Y. Tay, M. Dehghani, D. Bahri, and D. Metzler, "Efficient Transformers: A Survey," *ACM Comput. Surv.*, vol. 55, no. 6, 2023, doi: 10.1145/3530811.

[14] K. Han *et al.*, "A Survey on Vision Transformer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, 2023, doi: 10.1109/TPAMI.2022.3152247.

[15] A. Radford *et al.*, "Learning transferable visual models from natural language supervision," Feb. 2021.

[16] K. Weiss, T. M. Khoshgoftaar, and D. D. Wang, "A survey of transfer learning," *J. Big Data*, vol. 3, no. 1, 2016, doi: 10.1186/s40537-016-0043-6.

[17] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, 2011.

[18] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf. Process. Manag.*, vol. 45, no. 4, pp. 427–437, Jul. 2009, doi: 10.1016/j.ipm.2009.03.002.