Exploring robustness of image captioning in visual place recognition under appearance shifts

PEPIJN MEIJER, University of Twente, The Netherlands

Visual place recognition is currently one of the most important problems faced in the field of computer vision. It is the process of identifying the location of a given image and retrieving images captured at the same place. It is an essential component in the navigation of mobile robots, visual question answering, and autonomous driving. It is crucial that these models perform image retrieval tasks successfully in different weather conditions. In this study, we investigate the use of a captioning step within visual place recognition with the goal of improving the resistance of visual place recognition to differences in the query image. We specifically look at visual place recognition at a lower level by using a pipeline which emits the retrieval step and outputs image encodings, which would be used for image retrieval in an actual visual place recognition pipeline. We implement a pipeline using the ExpansionNet LLM to caption the image, the CLIP VLM to encode the image and the caption of the image. By introducing corruptions in the query image, we test the effectiveness of a captioning step in the pipeline. We find that with a caption, the results show a consistent 40% increase in resistance to corruption.

1 INTRODUCTION

Visual place recognition (VPR) is a task in computer vision which aims to retrieve images from a database which are geographically located in the same place as a given reference image. It is an essential component in the navigation of mobile robots [9], visual question answering [2], and autonomous driving [6, 17].

To this day, most VPR approaches are environment- and task-specific. While they perform great in controlled environments, their performance lacks in uncontrolled environments [11]. However, it remains imperative that a VPR system is able to recognise a location despite these uncontrolled environments; this is why Gosa [7] experimented with how different VPR pipelines were affected when corruptions or changes [4] were added to the input image. These corruptions are categorised into two types of corruption. The first type of corruption is Short-Term Corruptions, such as camera blur or image transformation. The second type of corruption is Long-Term Corruptions, consisting of weather or seasonal variations of the location.

We propose a pipeline using the work Gosa [7], which works by first generating captions for the input image using some model based on a large language model (LLM). Then the captions are encoded using a vision language model (VLM). These encodings are then combined with an encoding of the input image using the same VLM to obtain a more comprehensive embedding. We do this to gain insight into the use of a captioning step in the creation of an image embedding, to gain insight into the use of captioning within an actual VPR pipeline.

Author's address: Pepijn Meijer, p.j.meijer@student.utwente.nl, University of Twente, P.O. Box 217, Enschede, The Netherlands, 7500AE.

TScIT 43, July 4, 2025, Enschede, The Netherlands

2 RELATED WORKS

2.1 Current State-of-the-Art

Previous VPR methods trained a model using image pairs labelled as being in the exact location or not. However, these binary labels don't account for visual cues and are merely based on geographical distance. This is why Leyva-Vallina, Strisciuglio, and Petkov [13] introduced a method, which uses camera metadata or 3D information associated with image pairs to approximate a degree of similarity (Graded similarity) between the two images. Image pairs labelled using graded similarity can be used to train a VPR network without complex pair mining. This graded similarity is embedded into a Generalised Contrastive Loss (GCL) function to train a VPR pipeline [15, 23]. In this pipeline, they use a fully convolutional backbone to translate images into a matrix-like embedding.

Most state-of-the-art techniques use NetVLAD [3, 22, 8]. These techniques heavily rely on local features, which have proved to be resistant to viewpoint changes, but easily fail under camera blur or weather changes [16]. [1, 19]

To this day, most VPR approaches are environment- and task-specific. While they perform great in controlled environments, their performance lacks in uncontrolled environments [11]. However, it remains imperative that a VPR system is able to recognise a location despite these uncontrolled environments, because pictures of the same location are rarely the same. These differences or corruptions could include long-term corruptions like day and night shift, seasonal changes, weather changes, but also short-term corruptions like image capture corruptions, or domain changes, like temporary road work, garbage bins or walking people. Gosa [7] aimed to find out which of these corruptions were relevant for VPR, and tested the performance of multiple state-of-the-art pipelines with these corruptions implemented. He found that for short-term corruptions, blurring effects had the most impact on the performance of the system. For long-term corruptions, he found that day/night shift had the most impact on the performance.

2.2 Natural Language Representation

In their work Lee and Myung [12] show that Natural Language Representation is very promising in the research field of VPR, being interpretable to natural language. Since then, language models have developed significantly; models like Llama and ExpansionNet v2 have been shown to outperform their predecessors [10, 21]. Llama is a collection of foundation language models trained on trillions of tokens. [21]. In April of 2025, Meta AI released the fourth version of Llama, claiming to be more efficient than GPT-4o and Gemini 2.0. [20].

ExpansionNet v2 is a model created explicitly for image captioning

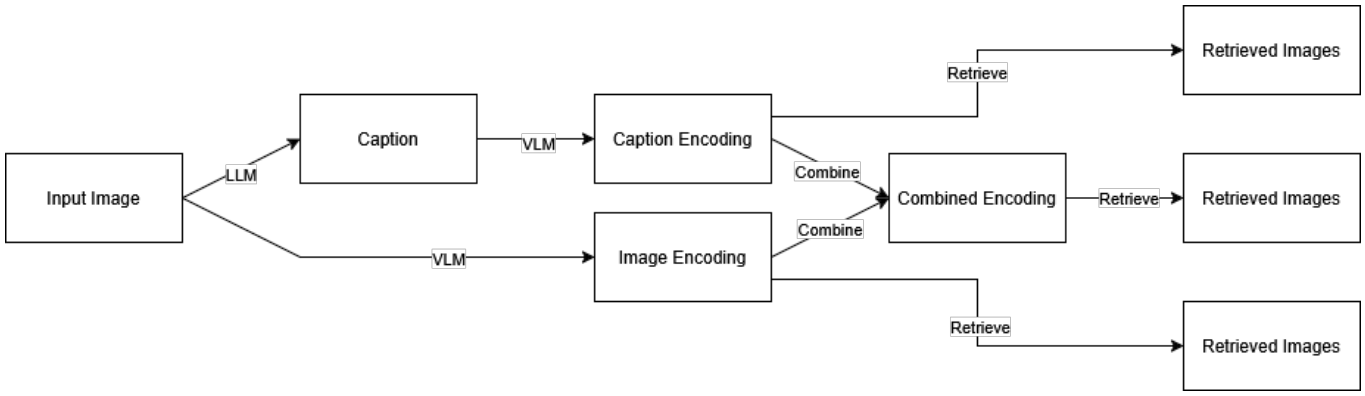

Fig. 1. VPR Pipeline

by Hu, Cavicchioli, and Capotondi [10]. This model utilises the Block Static Expansion method, performing forward and backwards expansion, which is effective and 6 times faster.

2.3 Vision Language Model

Large language models, such as LLaMa and GPT-4 have achieved remarkable success across a wide range of NLP tasks; however, as they continue to scale, challenges become apparent, like the finite supply of high-quality data and the limitations of their single-modality architectures. These limitations motivate the development of vision language models (VLMs). Li et al. [14] looked into the performance of multiple state-of-the-art VLMs, like CLIP, BLIP, Flamingo, GPT-4V, and Gemini. They lay out multiple benchmarks which can be used to test their performance.

Standard computer vision systems are trained to predict a fixed set of object categories. This restricts their use, as additional labelled data is needed to detect an object which the model was not trained on. This is why Radford et al. [18] proposed and implemented a new way of training computer vision systems, which can be used for predicting the category of an object which was not seen during training, also called zero-shot prediction. They jointly train both a text encoder and an image encoder. Before training, the images and captions are embedded into a vector space, then a matrix with the dot product of each image embedding against the text embeddings is created. After this, the encoder is trained to maximise the cosine similarity of the diagonal, which are the correct pairings, and minimise the cosine similarity of all other pairings, as seen in figure 2. They trained this model on 400 million (image, text) pairs. With different experiments, they show that CLIP (Contrastive Language-Image Pre-Training) is competitive with or outperforms the most popular training methods.

3 PROBLEM STATEMENT

This study tests the effectiveness of the study of Gosa [7] and the CLIP model [18] with the use of a simple non-complete VPR pipeline (Figure 1), with the goal of **testing whether image captions and text encodings have any impact to make VPR more resistant to changes**.

This is accomplished by first testing the similarity of the encodings

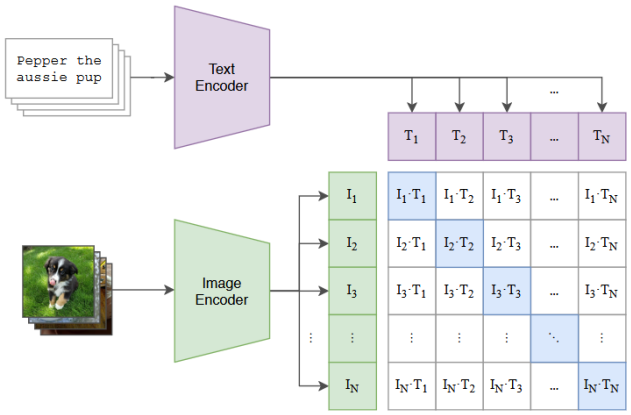

Fig. 2. CLIP Training Approach taken from the CLIP paper [18]

created using the clip model for different corruptions created using the work of [7] and comparing them against encodings made from image captions created by an LLM. The pipeline, as shown in Figure 1, consists of an LLM that generates a caption from the image, which is then passed through a VLM that encodes the image. In a complete VPR pipeline, these encodings would then be used to retrieve images from a database. The aforementioned steps and challenges lead to the following research question:

To what degree does adding an LLM-based captioning step, before encoding, to a VPR pipeline, affect the encodings created by the CLIP model when changes in the image are added?

With the following subquestions:

(1) How do changes to an image affect the encodings created by the CLIP model, compared to the uncorrupted image?
(2) How does the combined encoding of the caption of the image and the image itself compare to the combined encoding of the caption of the corrupted image and the corrupted image itself, when passed through the CLIP model?

4 METHODOLOGIES

In the study of Gosa [7], code has been implemented which adds corruptions to images; this code is used in the pipeline to corrupt images.

The pipeline consists of an LLM to caption the image and a VLM to encode the caption or the image.

In this study, we use the ExpansionNet V2 LLM by Hu, Cavicchioli, and Capotondi [10] to generate captions of the images.

The captions, generated by the ExpansionNet LLM, are then extracted into a compact feature vector and encoded using the CLIP vision language model. Alongside this, the original image is also extracted and encoded using CLIP. These two vectors, the caption encoding and the image encoding, are also combined, resulting in a total of three vectors: one for the caption encoding, one for the image encoding, and one for the combined encoding.

These encodings are then compared with the encodings of the corrupted images, created using the code from Gosa [7], which are also passed through the pipeline in the same way. This comparison is done by using cosine similarity in the vector space. Cosine similarity is a measure of similarity between two vectors. Cosine similarity is the cosine of the angle between the two vectors, calculated by using the following formula.

$$\frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} * \sqrt{\sum_{i=1}^{n} B_i^2}} \tag{1}$$

5 EXPERIMENTS AND RESULTS

5.1 Experimental Setup

For experimentation, images were taken from the san_francisco dataset [5]. This publicly available dataset has about 36,500 images taken in San Francisco. From this dataset, 2,586 images were corrupted using the code from the git repository from Gosa [7] his research, using five types of corruption: Defocus Blur, Motion Blur, Zoom Blur, Elastic Transform, and JPEG Compression, each with intensity levels ranging from 1 to 5, resulting in a total of 25 corrupted images for every original image. The experiments were run using Python on the HPC cluster of the University of Twente, using two of its GPUs. These images are then simultaneously processed with and without a caption.

5.1.1 Without Caption. For each of the 25 corrupted images and the original image, the image was passed through the OpenCLIP VLM to be encoded into a vector of the same size as the original image's encoding. This enables us to calculate the cosine similarity between every corrupted image encoding and the original image encoding, allowing us to evaluate the similarity between the corrupted image and the original image.

5.1.2 With Caption. For each of the 25 corrupted images and the original image, we simultaneously generate a caption for the image using the ExpansionNet LLM. This caption is then also passed through the OpenCLIP VLM to be encoded into a vector of the same size as the image vectors. For every image, the caption encoding is added to the image encoding and normalised, giving us an encoding of the same size as the encodings without a caption. These corrupted-image-plus-caption encodings are similarly compared to the encoding of the original-image-plus-caption using cosine similarity.

Comparing the cosine similarity of the images with and without captions enables us to gain insight into the effectiveness of a captioning step within a complete VPR pipeline.

With the cosine similarity scores for both with and without captions, we can compute one list for the cosine similarity scores without captions for every corruption type and intensity level, and another list for the cosine similarity scores with captions for every corruption type and intensity level.

5.2 Results

In this section, we will analyse and discuss the results, plotted in Appendix A. For every corruption type, there are three plots: the first plots the mean and median of each list of corruption types and intensity levels, to provide overall results and analyse trends across the entire dataset. The second plot shows a violin plot to examine each specific corruption intensity in more detail and to find trends in the density. The last plot shows the variance of each list of corruption types and intensity levels, as the violin plot analysis revealed a specific trend that would be better illustrated in a separate plot for variance.

These plots need two definitions to interpret the results correctly.
We define CS1 as: The cosine similarity score between the encoding of the original image and the encoding of the corrupted image.
We define CS2 as: The cosine similarity score between the encoding of the original image, added to the encoding of the caption of the original image and the encoding of the corrupted image, added to the encoding of the caption of the corrupted image.

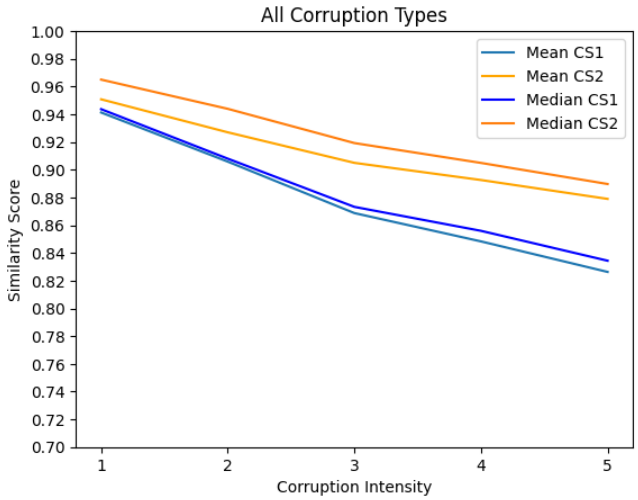

Fig. 3. Average and median cosine similarity score over different intensity levels with and without captions

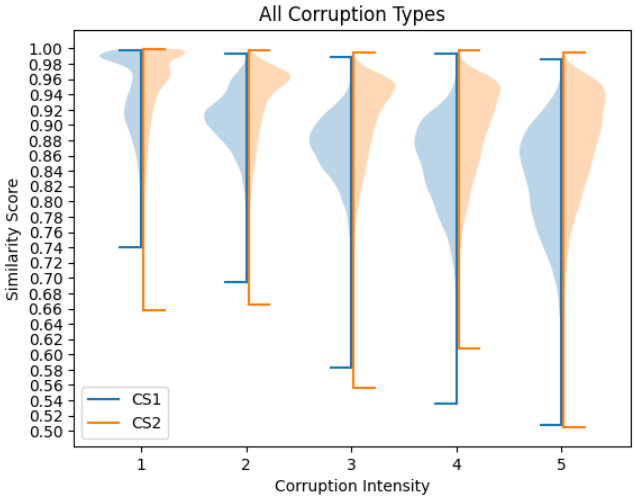

Fig. 4. Violin Plot showing the density of the cosine similarity score for images over different intensity levels with and without captions

The first plots above show some general results taken from all the cosine similarity scores for that intensity level. Figure 3 shows the overall average and median results. It shows a clear increase in the Similarity Score, with and without a caption, which is a promising result. The violin plot shows similar results, with most of the similarity scores with a caption being higher than the ones without a caption.

We will now present and discuss more detailed results for each corruption type, followed by an overview of the results.

5.2.1 Defocus Blur. A plot of the average and median cosine similarity is seen in Figure 5. It can be seen that with the caption, the cosine similarity decreases less as the intensity levels increase. It can be noted that the decrease in cosine similarity between intensity levels with the caption is approximately 40% lower compared to without the caption. It is also worth noting that for low corruption intensity, the average cosine similarity was higher without a caption than with one.
The violin plot seen in Figure 6 confirms the increased cosine similarity with the captions compared to without. It can be noted that the cosine similarities with captions have a denser tail below the average than the cosine similarities without, indicating a much higher variance, which can be confirmed with Figure 7, which shows the variance is consistently higher for the cosine similarities with captions, with approximately 0.00135 for every intensity.

5.2.2 Motion Blur. A plot for the average and median cosine similarity is seen in Figure 8. Similarly to defocus blur, it clearly shows that the average and median cosine similarity is higher with a caption than without. Notably, the decrease in cosine similarity between intensity levels appears to also be approximately 40% lower with a caption than without one. This pattern is only broken between corruption intensity levels three and four, where the decrease is higher with a caption than without. For motion blur, the average cosine similarity was consistently lower without a caption than with a caption. The median is also notably higher for the cosine similarity with a caption.
The violin plot seen in Figure 9 also shows very similar results to the plot for defocus blur, with the bulge being at approximately the same level as the mean. The density with caption also appears to have a tail towards the lower end, indicating that the variance is also higher, which is confirmed in the variance plot shown in Figure 10. This plot notably shows a constant increase in variance of approximately 0.00115 for every intensity level.

5.2.3 Zoom Blur. The zoom blur, seen in Figure 11, shows a very similar result. It shows that the average cosine similarity without a caption is consistently lower than with a caption, except for corruption intensity 1. This oddity is likely due to the tails in the violin plot seen in Figure 12, because there the bulge is slightly higher for the cosine similarities with the captions. This is also evident in the fact that the median remains consistently higher than the cosine similarities without a caption. Interestingly, again, the decrease in average cosine similarity between intensity levels is also approximately 40% lower with a caption compared to without. As seen in Figure 13, the variance in intensity levels is consistent with the other corruption types, as it shows a difference of approximately 0.00132 between the cosine similarities with and without caption.

5.2.4 Elastic Transform. The average and median cosine similarity for the corruption type elastic transform seen in Figure 14 are different from the rest. The corruption intensity does not look to have any effect on the cosine similarity, and only the presence of a corruption has an effect. The average and median cosine similarity with caption, however, show a consistent but higher cosine similarity than without caption.
The violin plot in Figure 15 also interestingly shows very similar results, with even the skew being very similar. Interestingly, unlike the other graphs, this graph shows an approximate 40% decrease in average cosine similarity between intensity levels. This is likely due to the very stable levels of cosine similarity not having a difference between corruption intensities that is pronounced enough to show a definite trend. The Figure 16 does, however, continue the trend of the cosine similarities with a caption to have a consistently higher variance compared to the cosine similarities without a caption, with this graph showing a consistent increase of around 0.00134.

5.2.5 JPEG Compression. JPEG Compression, seen in Figure 17, shows again similar results to the first three corruption types. The average and median cosine similarities are higher with the caption than without. The decrease in average cosine similarity between corruption intensity levels appears to be approximately 40% lower with a caption than without; however, this is not seen between intensity levels 1 and 2, where the average corruption intensity with a caption decreases more than the average corruption intensity without a caption.
The violin plot, seen in Figure 18, also shows similar results as the other corruption types with the cosine similarities with caption showing a tail below the average, the bulge for the cosine similarities with a caption also is consistently higher than the bulge for the

cosine similarities without a caption. The variance plot seen in Figure 19 also shows a consistent increase of approximately 0.00154.

5.2.6 Overview. The results reveal several clear and definite trends. In all graphs, the average and median of the cosine similarities with a caption are higher than those without a caption. This is only not seen at intensity level one in two graphs. The decrease in average cosine similarity between intensity levels with a caption is 40% lower than without a caption, suggesting that with a caption, the average cosine similarity becomes 40% more resistant to corruption intensity. There is a significant number of datapoints that do not share this trend and suggest that the average cosine similarity becomes less resistant to corruption intensity. The latter observations are, however, only seen in the cases where there is not a significant drop in cosine similarity between the two intensity levels.
In all the graphs, the density plot shows a definite skew towards the bottom for the cosine similarities with a caption; this skew is not present for the cosine similarities without a caption. The density graphs always show that the bulge is always higher for the cosine similarities with a caption than without a caption, even if the average cosine similarity for that entry is lower with a caption than without.
The variance plots show a clear trend that the variance of the cosine similarities with a caption is consistently approximately 0.00134 higher than the cosine similarities without a caption. However, for motion blur, the average variance increase is slightly lower, and for JPEG compression, the average variance increase is somewhat higher, but the variance plot definitely supports the observation that there is a constant increase in variance.

6 CONCLUSION

The study expands upon the previous works by partly introducing a new VPR pipeline, which includes a captioning step alongside a VLM model to increase resistance to corruptions, such as camera blur or image compression, but excludes a retrieval step, only outputting encodings of images. In the pipeline we use the ExpansionNet LLM to generate the caption and the OpenClip VLM to encode both the caption and the image into a vector of the same size. These captions and image vectors are then compared to those of the corrupted images passed through the same pipeline.

We find that the experiments show three clear trends.

6.1 Resistance of the image encoding to corruptions

When we include a captioning step in the pipeline, the results show that the pipeline is 40% more robust to corruptions over corruption intensity. This trend is not seen when the difference in cosine similarity between intensity levels is negligible; this could be a result of too low an amount of test images.

6.2 Skew in density results of images

When we include a captioning step in the pipeline, we see the density of all the results per corruption type per corruption intensity to be skewed towards the lower end. This shows that while on average the similarity is higher, there is less reliability of high results. However, the results at the lower end of the violin plot with a captioning step are similar to the average of the results without a captioning step. This skew is likely the result of the LLM, because the ExpansionNet v2 [10] LLM, used in the experiment, is not as advanced as LLMs like Llama [20]. This LLM generated captions such as: "Two cars driving down a street with a large building with a clock tower.", "A building with cars parked in front of a street.", and "A building with green umbrellas in front of a building." These captions, while accurate, could be more advanced and detailed.

6.3 Variance of experiment results

When we include a captioning step in the pipeline, the variance of results is constantly higher by approximately the same amount. Looking at the variance plots, we can visually see these results, as the plots with and without the captioning step appear almost identical, with the plot including the captioning step being consistently higher. This indicates that the skew observed in the previous point is highly consistent and likely due to the captions.

This allows us to answer our research questions

6.4 Answer to subquestion 1

Changes or corruptions in an image have a negative impact on the encoding similarity between the changed image and the unchanged images. With a small degree of corruption, this impact is already noticeable with a difference of approximately 5%, which only becomes more pronounced with a higher degree of corruption.

6.5 Answer to subquestion 2

With a captioning step, changes or corruption in an image show a negative impact on the encoding similarity between the changed image and the unchanged image. With a small degree of corruption, the caption has no effect and shows a similar difference of approximately 5%. This difference becomes more pronounced with a higher degree of corruption.

6.6 Answer to main research question

Adding a captioning step to a VPR pipeline is shown to be very promising. Adding a captioning step to the incomplete pipeline, without a retrieval step, shows a noticeable positive impact on the robustness of the encoding against corruption. The caption results in little difference for a low degree of corruption, but for higher degrees of corruption, the caption results in a 40% increase in resistance to corruption in the pipeline.

6.7 Future Work

6.7.1 Test cases. For the experiments, our resources were limited to 2 GPUs. This resulted in a limited number of test cases. For this reason, it was decided to limit test images to one of the datasets which Gosa [7] used in his research, and to limit the number of images. At some points during the analysis, the lower amount of datapoints was noticeable, which is why we recommend a follow-up research with more datasets, and more test images for each dataset.

6.7.2 Corruption Types. This research included five types of corruption: defocus blur, motion blur, zoom blur, elastic transform, and JPEG compression. These types were chosen to limit the number of

images that would need to be generated. However, these corruption types do not represent the long-term corruptions mentioned by Gosa [7]. In expanded research, corruption types, like snow, frost, fog, and others, would ideally be included.

6.7.3 Llama. Llama 4 is a recent version of the Llama LLM series, which was released in April of 2025. This LLM was initially planned to be used in the pipeline for this research; however, when conducting the experiments, the API was limited to the US, and the model itself was too large for our resources. An LLM like this could generate more specific captions by giving it more specific instructions.

6.7.4 Retrieval. During the experiments, we compared encodings of images; however, these encodings do not provide a perfect representation of the actual results of a VPR pipeline, which ultimately retrieves images from a database using the encoding. These encodings could still retrieve very different images, and to ensure the functionality of adding a captioning step, the retrieved images would also need to be compared to ensure the functionality of a retrieval step in an actual VPR pipeline.

ACKNOWLEDGMENTS

This research was done for the Bachelor's Assignment of the Technical Computer Science study at the University of Twente. We would like to give special thanks to Nicola Strisciuglio for supervising this research.

REFERENCES

[1] Amar Ali-bey, Brahim Chaib-draa, and Philippe Giguère. "MixVPR: Feature Mixing for Visual Place Recognition". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Jan. 2023, pp. 2998–3007.

[2] Stanislaw Antol et al. "VQA: Visual Question Answering". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. Dec. 2015.

[3] Relja Arandjelovic et al. "NetVLAD: CNN Architecture for Weakly Supervised Place Recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016.

[4] Giovanni Barbarani et al. *Are Local Features All You Need for Cross-Domain Visual Place Recognition?* 2023. arXiv: 2304.05887 [cs.CV]. URL: https://arxiv.org/abs/2304.05887.

[5] D. M. Chen et al. "City-scale landmark identification on mobile devices". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2011, pp. 737–744. DOI: 10.1109/CVPR.2011.5995610.

[6] Dzung Doan et al. "Scalable Place Recognition Under Appearance Change for Autonomous Driving". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 9318–9327. DOI: 10.1109/ICCV.2019.00941.

[7] V.I. Gosa. *Visual Place Recognition: Building an Evaluation Framework for Model Robustness*. July 2024. URL: http://essay.utwente.nl/100860/.

[8] Stephen Hausler et al. "Patch-NetVLAD: Multi-Scale Fusion of Locally-Global Descriptors for Place Recognition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2021, pp. 14141–14152.

[9] Michael Horst and Ralf Möller. "Visual Place Recognition for Autonomous Mobile Robots". In: *Robotics* 6.2 (2017). ISSN: 2218-6581. DOI: 10.3390/robotics6020009. URL: https://www.mdpi.com/2218-6581/6/2/9.

[10] Jia Cheng Hu, Roberto Cavicchioli, and Alessandro Capotondi. *Exploiting Multiple Sequence Lengths in Fast End to End Training for Image Captioning*. 2024. DOI: https://doi.org/10.1109/BigData59044.2023.10386812. arXiv: 2208.06551 [cs.CV]. URL: https://arxiv.org/abs/2208.06551.

[11] Nikhil Keetha et al. "AnyLoc: Towards Universal Visual Place Recognition". In: *IEEE Robotics and Automation Letters* 9.2 (2024), pp. 1286–1293. DOI: 10.1109/LRA.2023.3343602.

[12] Alex Junho Lee and Hyun Myung. "Natural Language Representation as Features for Place Recognition". In: *2022 19th International Conference on Ubiquitous Robots (UR)*. 2022, pp. 284–287. DOI: 10.1109/UR55393.2022.9826253.

[13] Maria Leyva-Vallina, Nicola Strisciuglio, and Nicolai Petkov. *Data-efficient Large Scale Place Recognition with Graded Similarity Supervision*. 2023. arXiv: 2303.11739 [cs.CV]. URL: https://arxiv.org/abs/2303.11739.

[14] Zongxia Li et al. *A Survey of State of the Art Large Vision Language Models: Alignment, Benchmark, Evaluations and Challenges*. 2025. arXiv: 2501.02189 [cs.CV]. URL: https://arxiv.org/abs/2501.02189.

[15] Feng Lu et al. *Towards Seamless Adaptation of Pre-trained Models for Visual Place Recognition*. 2024. arXiv: 2402.14505 [cs.CV]. URL: https://arxiv.org/abs/2402.14505.

[16] Carlo Masone and Barbara Caputo. "A Survey on Deep Visual Place Recognition". In: *IEEE Access* 9 (2021), pp. 19516–19547. DOI: 10.1109/ACCESS.2021.3054937.

[17] Olga Vysotska. "Visual Place Recognition in Changing Environments". PhD thesis. 2020. URL: https://hdl.handle.net/20.500.11811/8565.

[18] Alec Radford et al. *Learning Transferable Visual Models From Natural Language Supervision*. 2021. arXiv: 2103.00020 [cs.CV]. URL: https://arxiv.org/abs/2103.00020.

[19] Stefan Schubert et al. "Visual Place Recognition: A Tutorial [Tutorial]". In: *IEEE Robotics Automation Magazine* 31.3 (2024), pp. 139–153. DOI: 10.1109/MRA.2023.3310859.

[20] A blog post announcing the first models using Llama 4. 2025. URL: https://ai.meta.com/blog/llama-4-multimodal-intelligence/.

[21] Hugo Touvron et al. *LLaMA: Open and Efficient Foundation Language Models*. 2023. arXiv: 2302.13971 [cs.CL]. URL: https://arxiv.org/abs/2302.13971.

[22] Frederik Warburg et al. "Mapillary Street-Level Sequences: A Dataset for Lifelong Place Recognition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020.

[23] Sijie Zhu et al. *R^2Former: Unified Retrieval and Reranking Transformer for Place Recognition*. 2023. arXiv: 2304.03410 [cs.CV]. URL: https://arxiv.org/abs/2304.03410.

A GRAPHS AND PLOTS

In this appendix, we show the raw results as graphed from the data. Following is a small explanation of the terms used in the graphs.

We define CS1 as: The cosine similarity score between the encoding of the original image and the encoding of the corrupted image.

We define CS2 as: The cosine similarity score between the encoding of the original image, added to the encoding of the caption of the original image, and the encoding of the corrupted image, added to the encoding of the caption of the corrupted image.

A.1 Defocus Blur

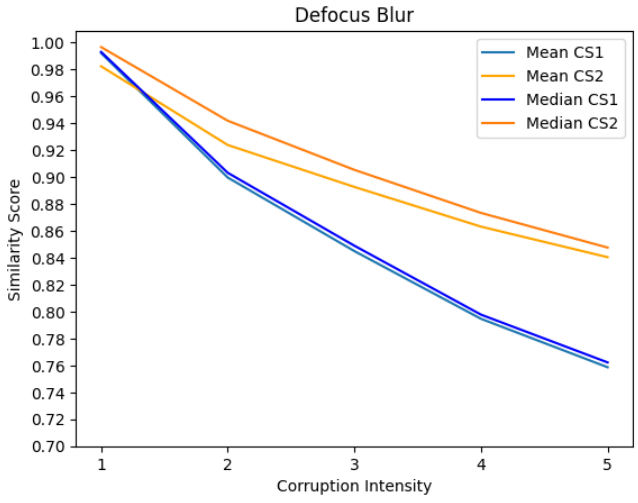

Fig. 5. Average and median cosine similarity score over different intensity levels with and without captions

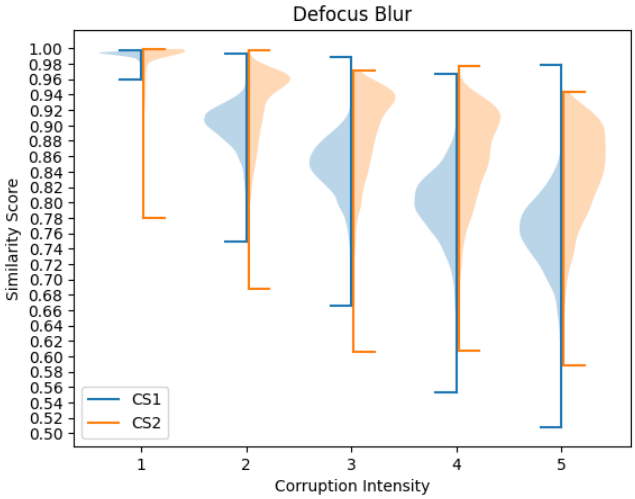

Fig. 6. Violin Plot showing the density of the cosine similarity score for images over different intensity levels with and without captions

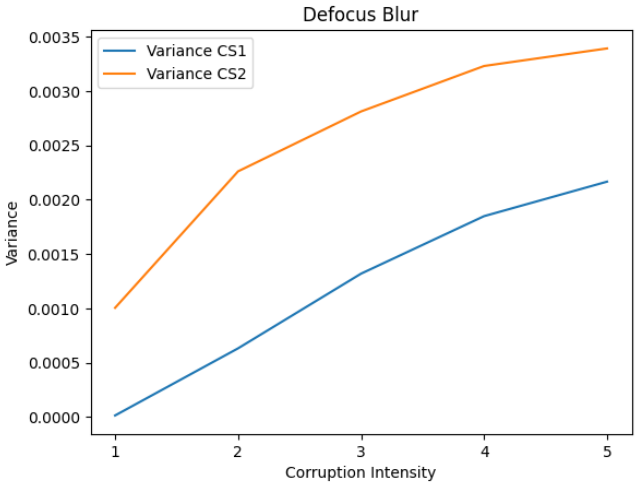

Fig. 7. Variance of the cosine similarity score over different intensity levels with and without captions

A.2 Motion Blur

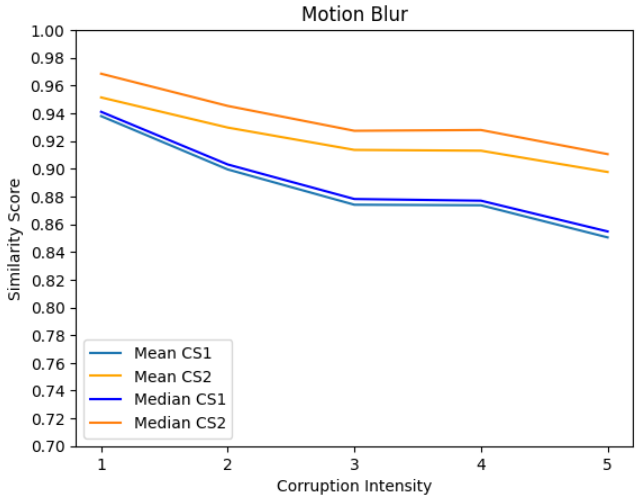

Fig. 8. Average and median cosine similarity score over different intensity levels with and without captions

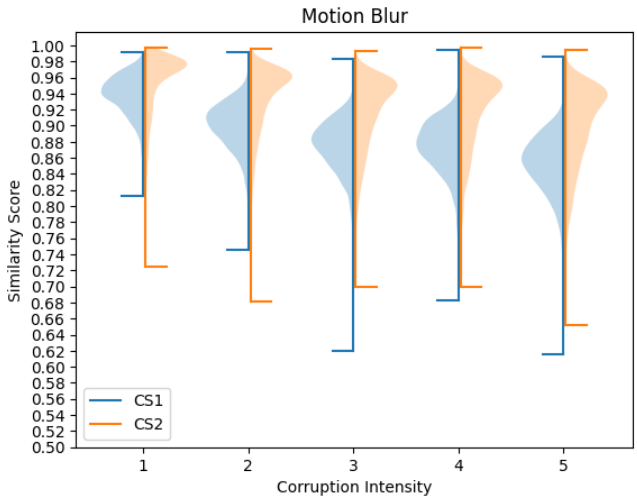

Fig. 9. Violin Plot showing the density of the cosine similarity score for images over different intensity levels with and without captions

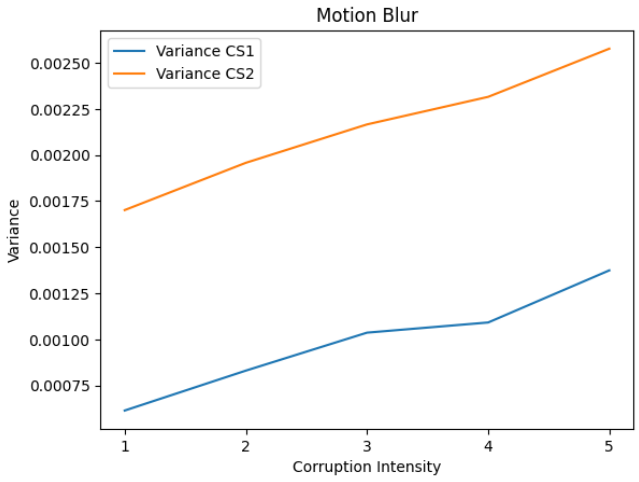

Fig. 10. Variance of the cosine similarity score over different intensity levels with and without captions

A.3 Zoom Blur

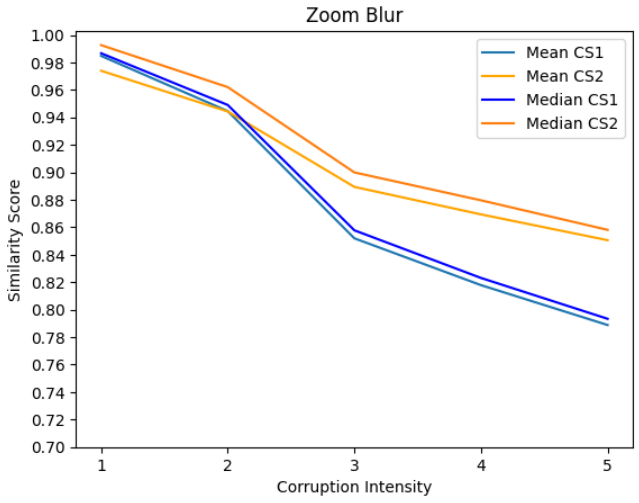

Fig. 11. Average and median cosine similarity score over different intensity levels with and without captions

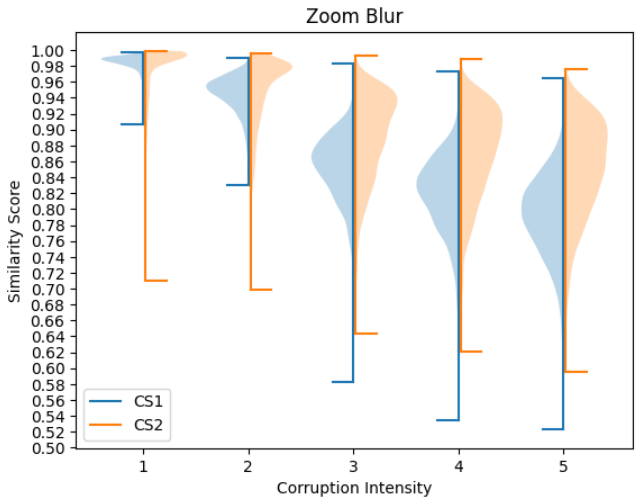

Fig. 12. Violin Plot showing the density of the cosine similarity score for images over different intensity levels with and without captions

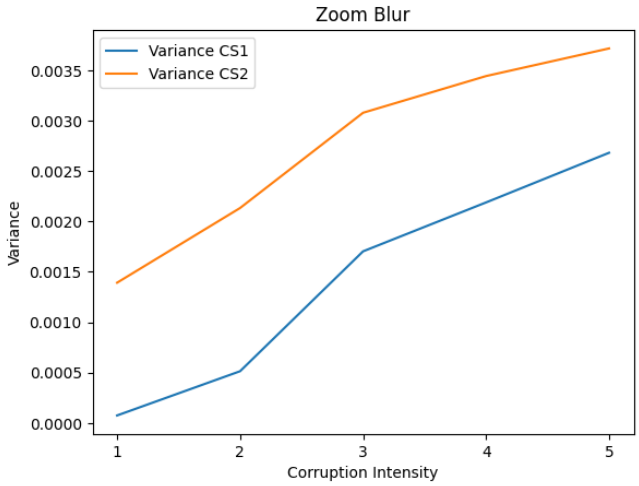

Fig. 13. Variance of the cosine similarity score over different intensity levels with and without captions

A.4 Elastic Transform

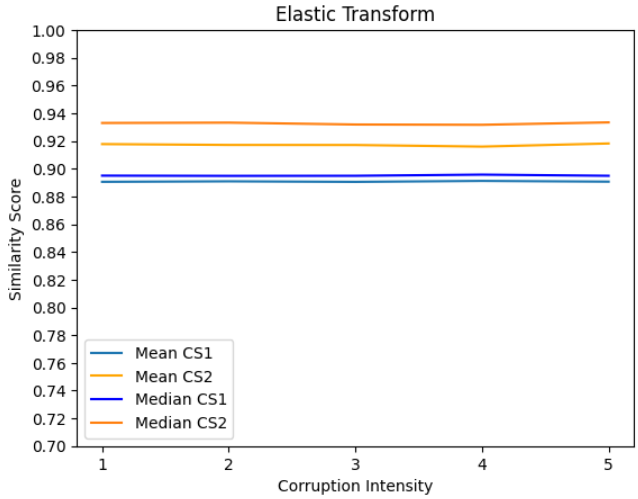

Fig. 14. Average and median cosine similarity score over different intensity levels with and without captions

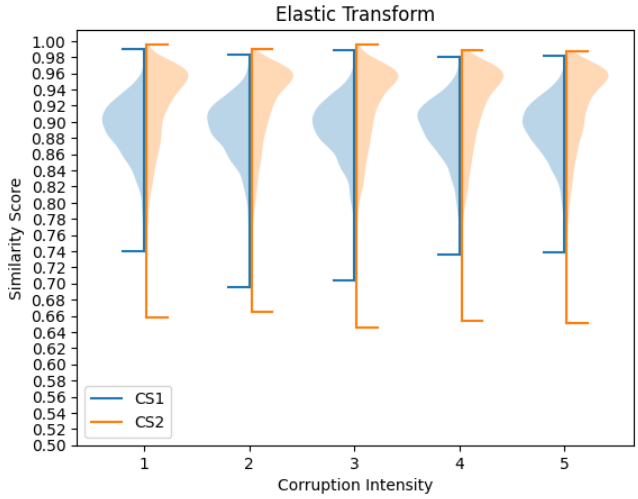

Fig. 15. Violin Plot showing the density of the cosine similarity score for images over different intensity levels with and without captions

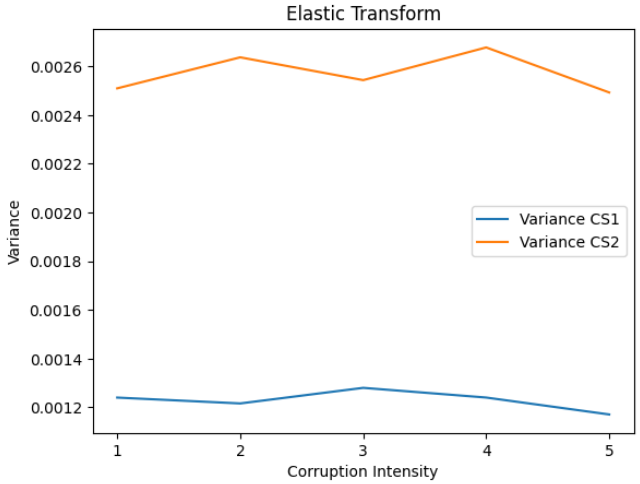

Fig. 16. Variance of the cosine similarity score over different intensity levels with and without captions

A.5 JPEG Compression

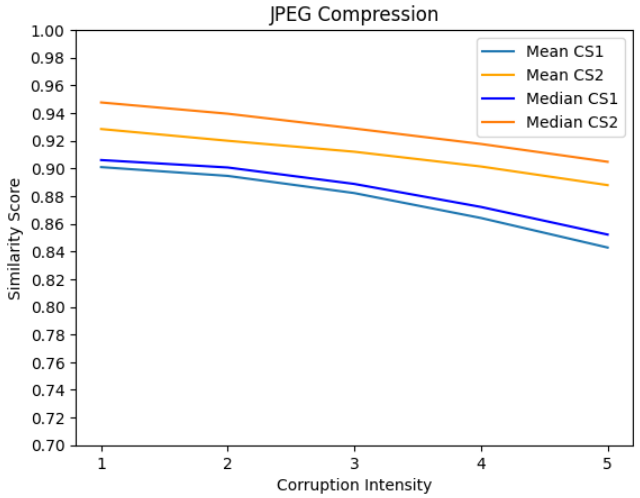

Fig. 17. Average and median cosine similarity score over different intensity levels with and without captions

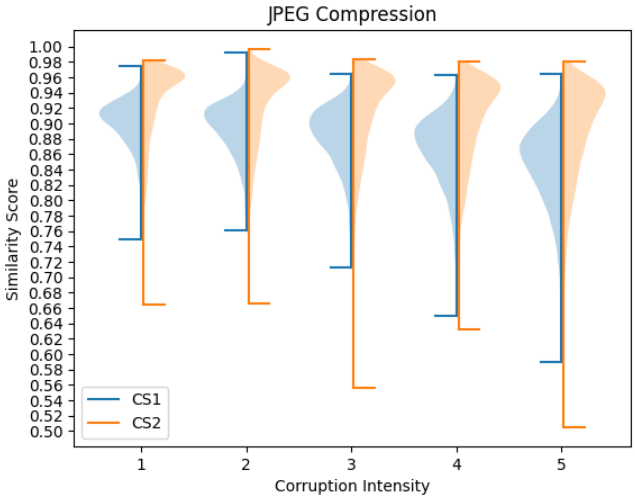

Fig. 18. Violin Plot showing the density of the cosine similarity score for images over different intensity levels with and without captions

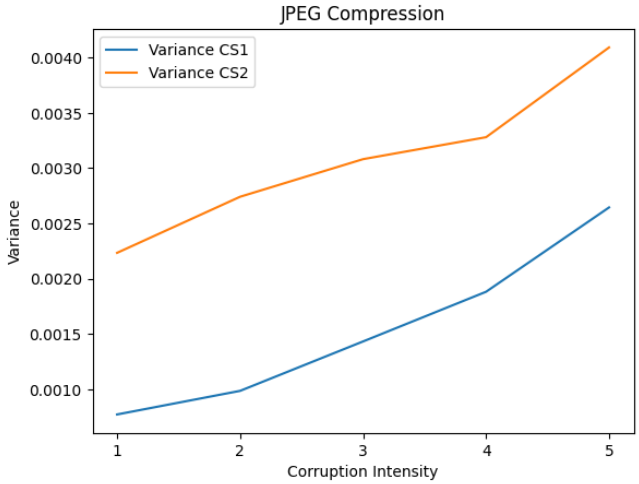

Fig. 19. Variance of the cosine similarity score over different intensity levels with and without captions

B AI DISCLOSURE

During this research, Grammarly was used to check spelling for the final report.