Benchmarking flow for testing sentiment coherence for information transfer models assessment

A. V. CHIROȘCA[1,*], G. CHIROȘCA[2,*], S. G. MUȘAT[2]
[1] *Faculty of Physics, University of Bucharest, Atomiștilor 405, 077125, Magurele, Ilfov, Romania*
[2]*National Institute of Research and Development for Optoelectronics INOE 2000, 077125, Măgurele, Ilfov,, Romania*

During this work we developed a benchmarking flow for subtle elements propagation through standard machine learning application. We started from the Stable Diffusion image generation models, one of the most active and innovative tools for artistic creation. While we focus on sentiment coherence by measuring the way image generation and image captioning models relate to this information. The benchmarking system incorporates a custom model that analyzes textual input (using Recurrent Neural Networks) providing text classification for the six primary emotional states: joy, sadness, love, anger, fear, and surprise. The classified emotion is used to construct a semantically rich textual prompt, which conditions the generative model to produce imagery aligned with the affective context of the input. The resulting visual outputs aim to faithfully encapsulate the emotional nuances conveyed in the source text. While working on-premises, we also compared the results with LLM (Large Language Model) to better grasp the capabilities of diffusion models in image generation. Then the produced image is fed through an image captioner model (Efficient Net B0) that once again produces a text caption that is analyzed through the sentiment analysis network. This approach shows that sentiments are hard to grasp by modern machine learning models providing a coherence score of 21%, the rest of the results being, most likely provided by model hallucinations. This approach demonstrates potential for a flexible application that can be used in many fields, providing feedback towards the model coherence (as a benchmark tool) for reducing hallucinations thus allowing model applications in fields such as digital art creation, art-based therapeutic interventions, affective computing, and the design of emotionally responsive human-computer interfaces.

(Received October 6, 2025; accepted December 4, 2025)

Keywords: AI art, Generative AI, Stable diffusion, Diffusion models, LLM, Sentiment analysis, Machine learning, Benchmarking, AI and creativity

1. Introduction

Newer integrated techniques [1] allow for the extraction of molecular level information it relays upon standard image analysis. Such techniques are often employed for art elements with respect to their authors, schools and other specific, physical characteristics extracted using all sorts of imaging techniques [1-4]. While these techniques are focused upon the substrate or the physical content of the artwork, one often less considered aspect is related to the message that it transmits as it's more subjective than quantitative or qualitative methods.

In order to capture subjective elements, the newest discoveries in Generative Artificial Intelligence (GenAI) [5, 6] techniques can provide a valuable, automated tool for such methods with a wide variety of parameters. This approach can be extended for other topics by inserting the topic of interest within the training flow. Our approach allows for the interpretation applied through machine learning algorithms allowing for digital access to information through technological advances [7]. For synthetic data generation we apply specially trained models like [8, 9].

While the traditional approach uses Generative Adversarial Networks (GAN's) initially proposed in 2014 [10] by the means of unsupervised learning training techniques its main purpose was to distinguish fake images from real ones [11] while the Least Squares approach for such algorithms provide reasonable results [12] they were limited in terms of interpretability and often provide inaccurate results from processes known under the name hallucinations. Within the recent years, their applications were limited to image reconstruction [13,14] and text generation [15] thus enabling multimodal analysis [16,17] allowing for correct context assessment. Such approach enabled us to construct, using the available Large Language Model servers a training dataset for our problem. By integrating such LLM's greatly reduces the time needed for training data assessment and calibration.

While GAN's presented interesting results we obtained more accurate models using the Latent Diffusion Method [18] where the latent space is defined using a Deep Recurrent Neural Network block.

This research aims to create a stringent benchmarking framework for assessing the propagation and impact of subtle affective data within common machine learning models.

2. Experimental

The experimental approach towards an image analysis framework containing of multiple models trained on different data sets we are estimating the model coherence after the message has pass through different media types. The overall data flow starts with two textual components, a prompt and a hint. The prompt contains the description for the image to be generated while the hint contains formal data (e.g. realistic, with humans, 1k) and output constraints. This information is fed to the generation block that output a generated image based upon the received instructions (the generator in Fig. 1). The generated image is then fed into the captioner model that generates a text based upon the image seen. The captioner output is then fed to the classifier (two classifier instances) are available performing text classification for both the prompt and the captioner results.

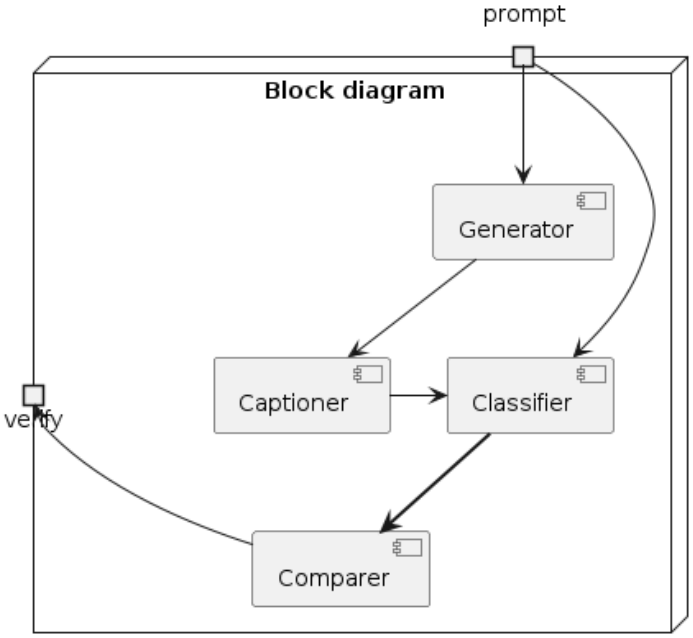

Fig. 1. Overall components block diagram

The results are then compared and generates a confusion matrix, the basis for our analysis.

The actual methodology implementation is based upon modular scripts developed using the Python 3.10 [19] language while the data procession workflow makes use of state-of-the-art libraries like Scikit-learn version 1.6 [20] for dataset manipulation including tokenization, Numpy version 2.3.0 [21] for numeric data handling and array manipulation and Pandas version 2.2.3 [22] for Data frame construction. The Machine Learning frameworks is TensorFlow v. [23].

2.1. The generator block

Our approach focuses upon the creation of a latent space with its specific higher dimensionality representation [18] implementing a auto-regressive network that focuses upon the correlation between the current period with respect to the latent result hat reverses the entanglement process of the Gaussian noise with the base image, thus generating a new image based on the received data. While the high-resolution data requires extensive processing power, we reduce the image size to a manageable dimension while, during the output, we employ the inversion technique [24] from denoising terminology in order to rescale the image to its original size.

This pipeline consists of three distinct stages, the text encoding phase where the user input gets evaluated, the diffusion component where the specific latent space is evaluated and the image decoding element.

For the encoder component, the Contrastive Language Image Preprocessing (CLIP) architecture [25] consists of a transformer [26] network that is specialized in written language recognition. For our case we are using cosine similarity for tensor representation assessment while we only keep the encoder part as an input condition for the elements down the pipeline [27].

The second component consists of the diffusion phase where we employ a U-net architecture [28, 29] that is an established method for image analysis (also a transformer), that performs image segmentation. This stage applies a gaussian blur filter altering the image and, on the decoder, side reconstructs the image while taking into account the output of the CLIP stage (the encoder component) thus leaving us with an augmented image (Fig. 2).

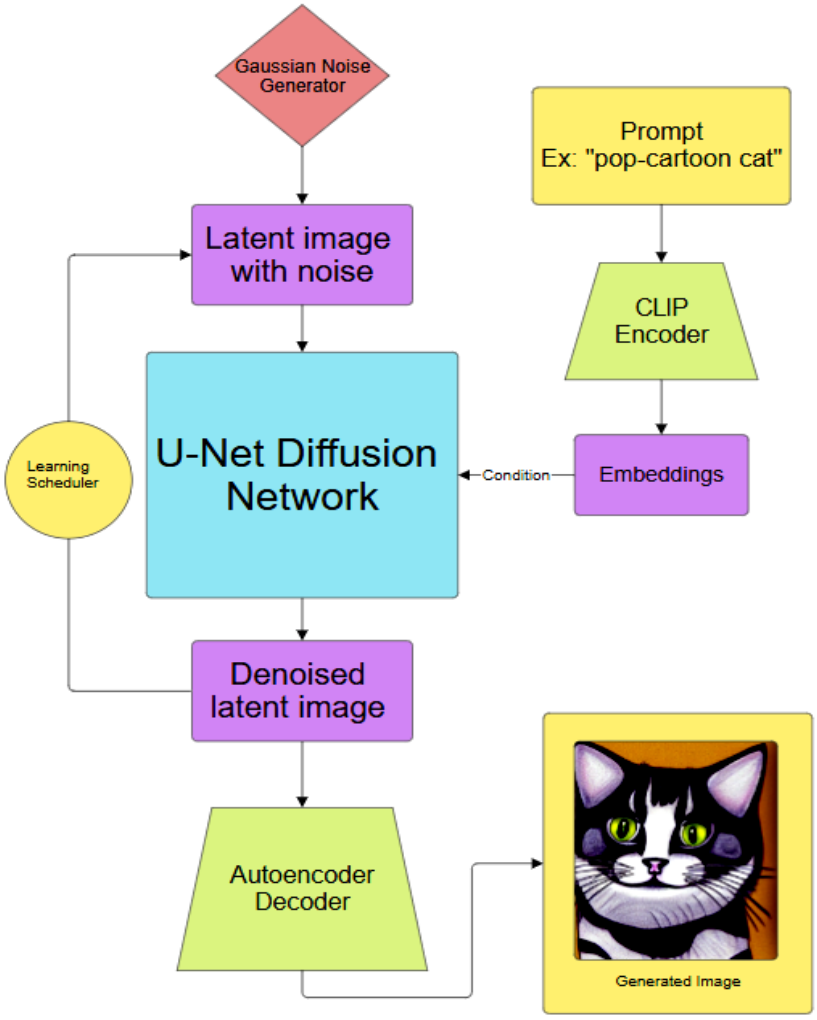

Fig. 2. The input query text "pop-cartoon cat" is prompted to the Latent Diffusion model. The convolutional network is used in the Gaussian denoising process which is regulated by the extracted text embedding of the original input query using a transformer encoder. The process is regulated by a learning scheduler of 50 steps, after which the autoencoder decoder is used to reconstruct the image is high resolution (colour online)

Such architecture allows for image generation with minimal associated computing power. There is still a need to optimize the computing environment by using the JAX infrastructure thus minimizing both the computing time and the required resources.

Once the architecture is settled, we continue to train our workflow. While our work focuses on sentiment analysis, we use a publicly available dataset [30] containing of 393822 entries mapped for 6 classes (joy, sadness, anger, love, fear and surprise (Fig.3). As this dataset is unbalanced, the classifier network needs to be unbiased.

The dataset [30] has punctuation removed and in lowercase. We will use 30% of the dataset, upon shuffle for testing (15%) and validation (15%) purposes while 70% of the dataset is for training.

2.1.2. Performance evaluation

The main purpose for this algorithm is to generate digital images (assets) with high accuracy based upon the user requirements including tone and words from a prompt interface while keeping certain aesthetic aspects as instructed. Sometimes the received instructions from the prompt provides a more cluttered image but the overall images are consistent for our case.

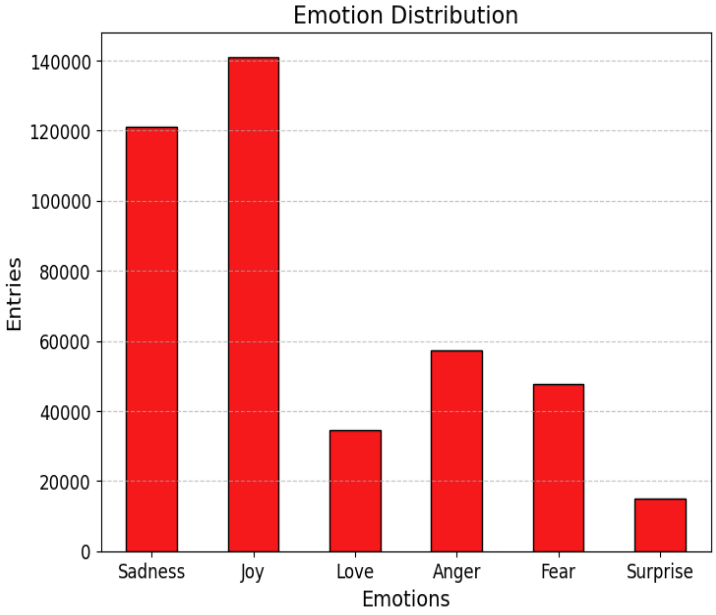

Fig. 3. Dataset record count – the first two classes are more prominent than all other (colour online)

Traditionally such use case involves Large Language Models like Gemini 2.0 Flash [31] the results for such systems lack in consistence and interpretability usually mixing up elements and confusing the user. The LLM large architecture (autoencoder [32] or transformer based) provides generic generation capabilities [33] unlike the proposed diffusion model that focuses on specific images and text input (prompt) rather than general purpose inputs.

On the other hand, using the U-Net architecture requires images to have smaller resolution leading to a reduction in entropy and simply scaling it up leads to generation of artifacts. In order to overcome this issue, we are implementing a denoising algorithm [34].

For model performance assessment we are comparing the model results with the ones provided by an established Large Language Model [35] where the same prompt was given to both models. The results are presented in Figs. 8-11.

2.2. The captioner block

Using standard available tools, the image captioner block (Fig.4) is responsible for converting images to text thus enabling us to evaluate the captioned text with respect to the initial generated image.

For this block we start from the Efficient Net B0 [36] architecture without any frozen layers, thus all layers are adjusted during the training phase. While the training phase requires images to be correctly labeled, we will use the Kaggle Flikr8k Dataset [37]. The dataset was split into 6114 (80%) training images and 1529 (20%) validation. The training step takes approximatively 2 minutes on A2000 16 Gb GRAM and the resulted accuracy (with respect to expert annotations is: 42%).

The model uses a Fully Connected Network transformer architecture that employs an positional embedding latent space of dimension 512 with a queue length (sequence length) of 25 covering the entire token vector space. The input phase consists of the EfficientNetB0 block while both the encoder and the decoder use Layer Normalization to mitigate gradient issues. Both transformer components (the encoder and the decoder) take advantage of the Multihead Attention with one head for the encoder part and two heads for the decoder part.

While the generation of caption text is not exact – and it should not be exact as the caption is a subjective term – an similarity of 42% with the expert captioner is considered to be an adequate one.

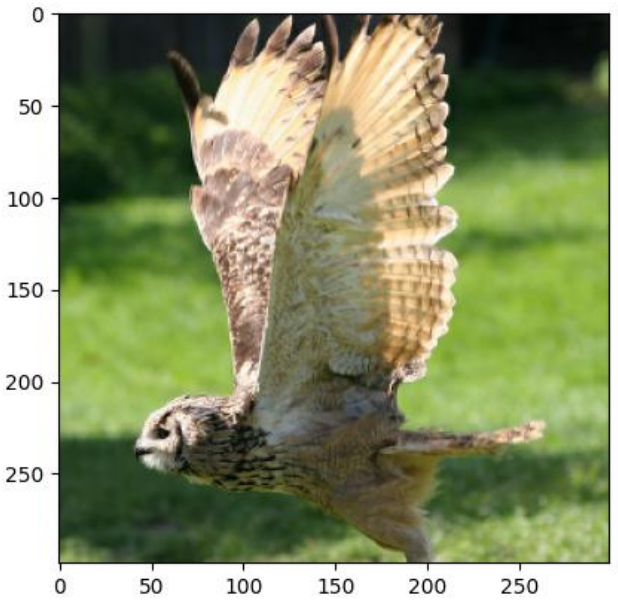

Fig. 4. Image captioned as "a bird is flying through the air", image is from the validation dataset for the captioner model [37] (colour online)

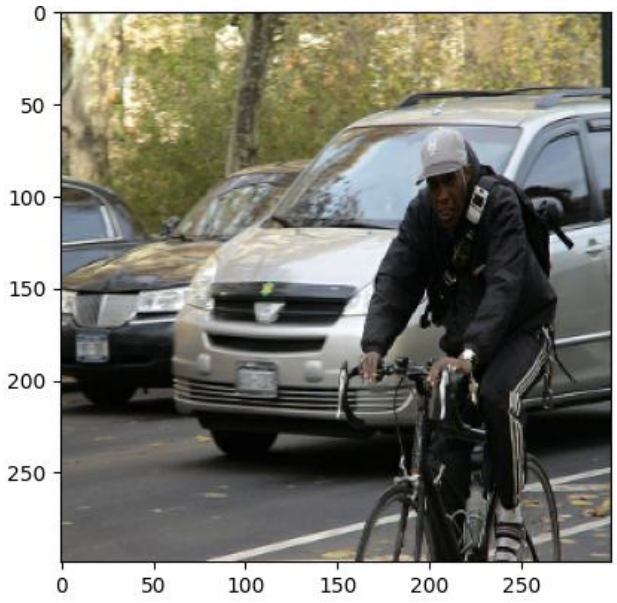

Fig. 5. An image captioned as "a man riding a bicycle down a road" by the captioner model. Image is from the Flikr8k Kaggle Dataset [37] (colour online)

While the model can be improved to provide better captions for the input images (Fig.5), our aim is to evaluate the Application for Latent Diffusion Models for Automated Digital Asset Analysis with a focus on sentiment assessment sentiment coherence level between the input phrase sentiment and the output.

2.3. The classifier block

The latent space does not need, at least for our approach, a very large size and its main purpose is to encode the user input and provide the reconstruction phase with accurate context. In order to perform input prompt analysis, we need Recurrent Neural Network architecture. In our approach we rely upon the standard Long Short-Term Memory (LSTM) [38] layers. The overall architecture is presented in Fig. 6.

Model: "sequential"

Layer (type)	Output Shape	Param
embedding (Embedding)	(None, 100, 64)	640,000
lstm (LSTM)	(None, 100, 128)	98,816
batch_normalization (BatchNormalization)	(None, 100, 128)	512
lstm_1 (LSTM)	(None, 64)	49,508
dense (Dense)	(None, 128)	8,320
dense_1 (Dense)	(None, 6)	774

Total params: 2,392,980 (9.13 MB)

Trainable params: 797,574 (3.04 MB)

Non-trainable params: 256 (1.00 KB)

Optimizer params: 1,595,150 (6.09 MB)

Fig. 6. Latent space LSTM deep neural network (colour online)

The best model training parameters are estimated at 10^{-3} learning rate with the Adam [39] optimizer while using the categorical cross entropy loss function with a batch size of 32. The training process uses callbacks for stopping the training process where no improvement is recorded or when overfitting is detected. Such events were

detected at a very small epoch count (only 28) leading towards a training score of 94.38% while the validation (the data is not seen the model during training) is of 94.18%. The recorded validation loss value is less than 0.1%.

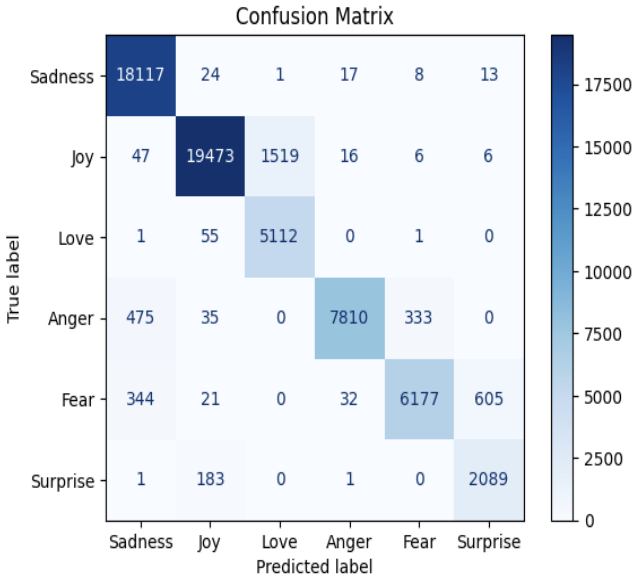

Fig. 7. Confusion matrix for the synthetic data generation model for sentiment analysis (colour online)

The confusion matrix from Fig. 7 clearly shows the models coherence towards the sentiment correlation. While the first diagonal shows correctly predicted values there is a similarity between sadness, anger and fear while joy is correlated with surprise or love. The interesting miss result is for surprise where the closest emotion is fear thus our model understands the human behavior and the relationships between sentiments.

3. Results

While machine learning models have several applications in Cultural Heritage [40,41], it often fails to capture the subjective facts that makes the element more capturing to the audience. Within this work we are focusing upon the sentiment coherence transmission between several AI/DL models. This coherence provides an interesting benchmark for the involved models [42].

The coherence is evaluated with the sentiment detection algorithm starting with the caption text going through the sentiment analysis block and being classified. Then the same text is fed to the image generation model (Generator) and the resulting image is the fed to the captioner becoming once again text. This caption text is then fed to the sentiment classifier and its class is then compared with the initial classification. If both classes are identical then we have a successful coherence test and when they differ the result is not successful.

There is a special case when the generator model is providing images containing text or are having clear hallucination effects (the generate image has no relation with the input phrase. Such cases (less than 15%) are discarded from the analysis. Even in this case, interesting results are obtained.

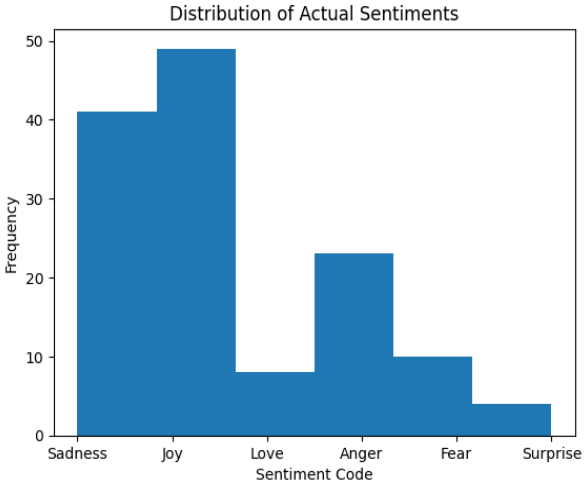

Fig. 8. Sentiment coherence input dataset structure (colour online)

By performing this flow on a reduced, not previously seen by the model we can evaluate the sentiment coherence degree within our process. In order to perform this, we selected, from the validation set, a subset of 135 prompts. The sentiment distribution for these prompts is presented in Fig. 8.

By taking these prompts and feeding through the benchmarking process we obtained a second distribution (presented in Fig. 7) containing the inferred sentiment values.

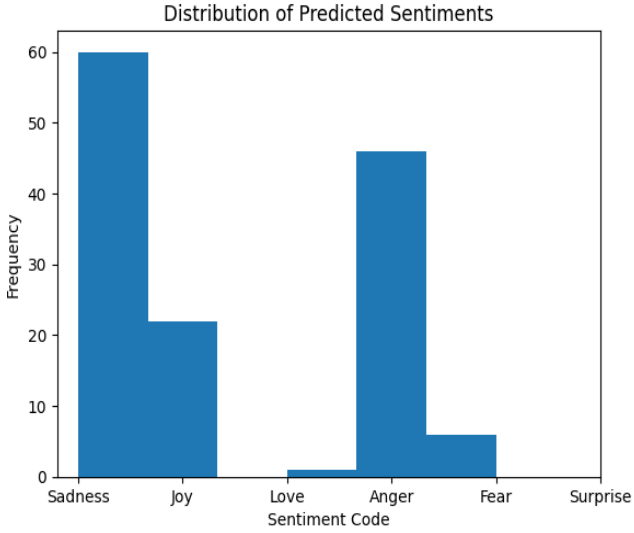

Fig. 9. Coherence evaluation result histogram (colour online)

By analyzing these two figures we determined that the captioning model provides a biased result towards negative sentiments like Anger and Sadness. While this seems to be related to the classifier model, the confusion

matrix presented in Fig. 9 contradicts this as it provides more than optimal classification results.

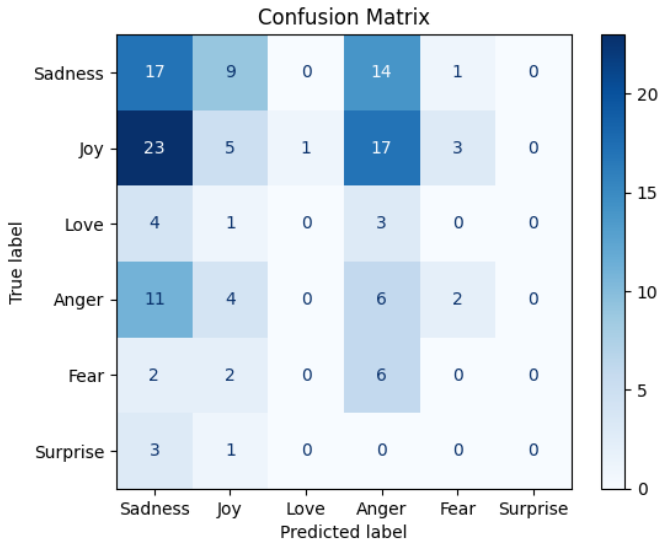

Fig. 10. Coherence evaluation confusion matrix (colour online)

The confusion matrix from Fig. 7 shows that the results are accurate only for 21% of the sentiment coherence test dataset. This subset, presented in Fig. 10, shows that the most correctly identified sentiment is Sadness (index 0) encountered in 50% of the cases, followed by Anger 21% and then by Joy 18%. Other less significant values occupy the rest of 11% of the correctly identified sentiment (sentiment coherence hit).

While the model training results provided high level of confidence for the models within the benchmarking system, the overall results show that the sentiment coherence level transferred between text – image – text is less confident than in the case of directly applied tools like [38], [41], [43] and [44]. Fig.11 displays the distribution of correctly predicted sentiments.

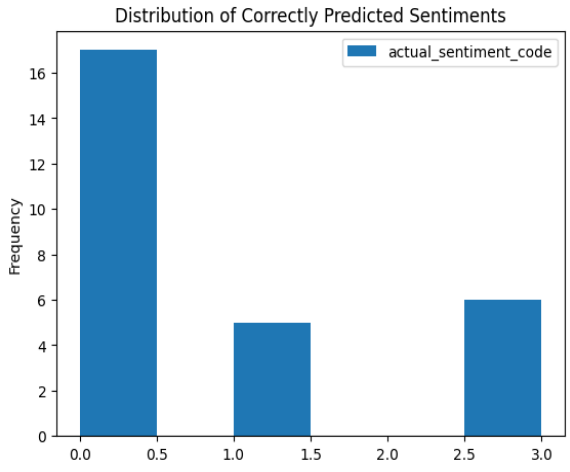

Fig. 11. Correctly predicted sentiments (0 - Sad, 1- Joy, 2-Love, 3-Anger, 4-Fear, 5-Surprise) (colour online)

4. Conclusion

This work was performed for sustaining the process of benchmarking machine learning/deep learning technologies when dealing with subjective information (as opposite to mathematical concepts usually found in physicochemical, scientific analysis). We focused upon one of the most challenging (subtle) aspect related to the sentiment propagation in machine learning models. Within this process we interpreted the text to image and image to text transformer frameworks, starting from the Stable Diffusion model for image generation (that performed reasonably on RTX 4090 GPU's and took a long time for RTX A2000 and similar with the same amount of GPU RAM). While all models are implemented locally, tests were developed using standard cloud available tooling.

For the image captioning an award-winning model was selected (Efficient Net B0) trained upon one of the most important image caption datasets publicly available.

While publicly aware models are available, they were quite limited in terms of sentiment detection (even with our large dataset from Kaggle) and we implemented a specific one.

During the data analysis step, we found that models were strongly biased towards negative strong sentiments while a shallow sentiment like surprise were completely ignored by the machine learning image captioner. This makes sense because surprise that occurs when we encounter an unexpected event or information. It can be positive, negative, or neutral, depending on the situation. While this sentiment is usually leading to stronger feelings like joy, fear, or anger it's possible to mismatch it.

The biggest issue we found that sadness, anger and joy are usually misplaced by either the captioner or the generator (Stable Diffusion) model, although locally trained it still generates a lot of images with its original training. This process can also identify and provide accurate assessment for the hallucination process for machine learning.

The benchmarking process provides a valuable tool for the assessment of machine learning models focus upon subtle aspects of the Computer Vision while coherence within machine learning field is becoming a key factor in the last years.

Such algorithms provide a basis for the development of hallucination resilience extending the generative model for a wider range of applications leading to enhance the robustness and the confidence level of generative model-based applications.

Inter-Model Benchmarking: Expanding the comparison (currently against LLMs) to include other generative image models (e.g., Midjourney, DALL-E 3) to derive a broader conclusion regarding the affective coherence performance across different architectures. A prospective direction suggests that the workflow is sufficiently reliable to be integrated as a foundational tool within a digital art therapy application, thus demonstrating its potential to function as a digital mirror of the user's emotional state. An essential conclusion would be that the benchmarking methodology provides a framework for the

ethical auditing of generative models, ensuring that they neither propagate nor amplify harmful stereotypes or inappropriate content associated with specific emotions.

Acknowledgements

This work was carried out through the Core Program with the National Research Development and Innovation Plan 2022–2027, with the support of MCID, project no. PN 23 05/2023, contract 11N/2023, and iPhotocult—Intelligent Advanced Photonics Tools for Remote and/or on-site Monitoring of Cultural Heritage Monuments and Artefacts, project no. 101132448/2024.

References

[1] M. Dinu, L. Ghervase, L. C. Ratoiu, I. M. Cortea, L. M. Angheluta, A. M. Patrascu, C. M. Stancu, V. A. Cristea, J. Optoelectron. Adv. M. **26**(11-12), 512 (2024).
[2] J. Striber, R. Radvan, L. M. Angheluta, J. Optoelectron. Adv. M. **11**(11), 1815 (2009).
[3] L. Angheluta, A. Moldovan, R. Radvan, UPB Scientific Bulletin, Series A: Applied Mathematics and Physics **73**(4), 193 (2011).
[4] I. M. Cortea, A. Chiroşca, L. M. Angheluţă, G. Seriţan, ACM Journal **16**(2), 1 (2023).
[5] W. Hersh, K.F. Hollis, npj Digit. Med. **7**, 247 (2024).
[6] T. Zhou, X. Wu, Humanities and Social Sciences Communications **11**, 1083 (2024).
[7] S. Suri, Nature Computational Science **4**, 641 (2024).
[8] W. Y. Leong, Eng. Proc. **92**(1), 45 (2025).
[9] C. Avlonitou, E. Papadaki, Arts **14**(3), 52 (2025).
[10] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative Adversarial Networks **1406**, 2661 (2014).
[11] A. You, J. K. Kim, I. H. Ryu, T. K. Yoo, Eye and Vision **9**, 6 (2022).
[12] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, S. P. Smolley, IEEE International Conference on Computer Vision (ICCV) **1611**, 04076 (2017).
[13] D. Wang, C. Ma, S. Sun, Applied Sciences **13**(18), 10379 (2023).
[14] A. Asperti, G. Colasuonno, A. Guerra, Applied Sciences **13**(11), 6487 (2023).
[15] V. R. M. K. Gopu, M. Dunna, Journal of Imaging **10**(6), 139 (2024).
[16] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, arXiv **2005**, 14165 (2020).
[17] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, I. Sutskever, arXiv **2102**, 12092v2 (2021).
[18] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, B. Ommer, arXiv **2112**, 10752v2 (2022).
[19] Python Software Foundation, *Python*. Version 3.10. Python Software Foundation, 2023, https://www.python.org/
[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Journal of Machine Learning Research **12**, 2825 (2011).
[21] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, Travis E. Oliphant, Nature **585**, 357 (2020)
[22] W. McKinney, Proceedings of the 9th Python in Science Conference, 56 (2010).
[23] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, X. Zheng, arXiv **1605**, 08695v2 (2016).
[24] P. Esser, S. Kulal, A. Blattmann, R. Entezari, J. Muller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, F. Boesel, D. Podell, T. Dockhorn, Z. English, K. Lacey, A. Goodwin, Y. Marek, R. Rombach, arXiv **2403**, 03206v1 (2024)
[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, arXiv **1706**, 03762v (2023).
[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, arXiv **1706**, 03762v7 (2023).
[27] A. Radford, J.W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, I. Sutskever, arXiv **2103**, 00020v1 (2021).
[28] O. Ronneberger, P. Fischer, T. Brox, arXiv **1505**, 04597v1 (2015).
[29] W. Shi, J. Caballero, F. Huszar, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, Z. Wang, arXiv **1609**, 05158v (2016).
[30] N. Elgiriyewithana, Emotions, Kaggle, Dataset DOI: https://doi.org/10.34740/KAGGLE/DSV/7563141.
[31] Google, Gemini 2.0 Flash API, https://ai.google.dev/gemini-api/docs, 2024.
[32] D. Bank, N. Koenigstein, R. Giryes, Autoencoders **2003**, 05991 (2021).
[33] M. Watson, D. S. Sreepathihalli, F. Chollet,

M. Gorner, K. Sodhia, R. Sampath, T. Patel, H. Jin, N. Kovelamudi, G. Rasskin, S. Saadat, L. Wood, C. Qian, J. Bischof, I. Stenbit, A. Sharma, A. Mishra, C. V. Keras, N. L. P. Keras, Vision and Language Power-Ups **2405**, 20247 (2024).
[34] J. Ho, A. Jain, P. Abbeel, arXiv **2006**, 11239v (2020).
[35] F. Chollet, L. Wood, D. Gupta, High-performance Image Generation Using Stable Diffusion in KerasCV, https://www.tensorflow.org/tutorials/generative/generate_images_with_stable_diffusion.
[36] Aihua Zhou, Yujun Ma, Wanting Ji, Ming Zong, Pei Yang, Min Wu, Mingzhe Liu, Multimedia Systems **29**, 487 (2023).
[37] Kaggle Flikr8k Dataset, https://www.kaggle.com/datasets/adityajn105/flickr8k.
[38] S. Hochreiter, J. Schmidhuber, Neural Computation **9**(8), 1735 (1997).
[39] Diederik P. Kingma, Jimmy Ba, arXiv **1412**, 6980 (2014).
[40] G. Chiroșca, R. Rădvan, M. Pop, A. Chiroșca, Applied Sciences **15**(16), 9014 (2025).
[41] G. Chiroșca. R. Rădvan. S. Mușat, M. Pop, A. Chiroșca, Applied Sciences **15**(1), 212 (2025).
[42] M. Alikhani, B. Khalid, M. Stone, Front. Artif. Intell. **6**, 1048874 (2023).
[43] C. Dumitrescu, Optoelectron. Adv. Mat. **19**(7-8), 384 (2025).
[44] L.M. Angheluță, R. Rădvan, 27th Cipa International Symposium: Documenting the past for a better future **42-2**(W15), 101 (2019).

*Corresponding author: alecsandru.chirosca@unibuc.ro; gianina.chirosca@inoe.ro