

Real-Time Recognition of Unseen Destinations for Drone Delivery: A Multi-Modal and Generative Approach

Victoria Eugenia Vázquez Meza, Delia Irazú Hernández-Farías, Jose Martinez-Carranza*

Instituto Nacional de Astrofísica, Óptica y Electrónica,
Computer Science Department, Puebla,
Mexico

carranza@inaoep.mx

Abstract. We investigate the application of generative models to assist artificial agents, such as delivery drones or service robots, in visualising unfamiliar destinations solely based on textual descriptions. To this end, we propose a methodology aimed at comparing generated images from textual descriptions against images captured with a camera onboard a drone. Our methodology involves using clustering to organize thousands of generated images, thus enabling a rapid comparison. For this purpose, we employ visual embeddings such as CLIP to numerically represent the images and make them comparable at a semantic level. However, when two candidate images are similar in semantic terms, it is useful to assess which one more closely resembles the target image visually. Therefore, we also explore the use of visual features obtained with ResNet. Then, we combine them through a measure that scores the similarity between CLIP embeddings and ResNet features of the images to be compared. We achieve an accuracy of 0.80 with an average operation frequency of 18 Hz for online image processing, demonstrating the feasibility of our approach for recognizing previously unseen places for real-time drone applications.

Keywords. Place recognition, drone delivery, generative model, stable diffusion, CLIP.

1 Introduction

In recent years, the e-commerce sector has seen a significant increase in the number of packages requiring delivery each day, handled by postal services, shipping companies, or local delivery services. However, when Amazon launched its Amazon Prime Air project in 2013,

it inspired numerous companies and researchers to explore drone delivery, including its associated challenges [8].

Delivery services face logistical challenges, with one of the most costly being the efficient and cost-effective completion of deliveries from local warehouses to customers. This challenge is commonly referred to as the *last-mile delivery problem* [4]. Some of these challenges include scenarios such as urban design being disorganized, making it difficult for courier deliveries to locate the destination even when GPS is available; customers not being at home to receive the parcel; or couriers not being able to find the delivery location.

To address these challenges, a courier typically calls the user to ask for and receive information about the target destination. Alternatively, a picture of the place would greatly help the courier to visually seek out the destination. However, at a larger scale, storing each destination image would be costly. Moreover, receiving large volumes of visual information from customers can raise privacy concerns. Due to these challenges, especially in poorly designed urban areas, many delivery services already request a textual description to assist the courier in locating the destination, which is less memory-intensive than storing an image.

In previous works [5, 19], we introduced the idea of using textual descriptions (similar to how a courier consults descriptions provided by the customer) to identify unseen destinations successfully.

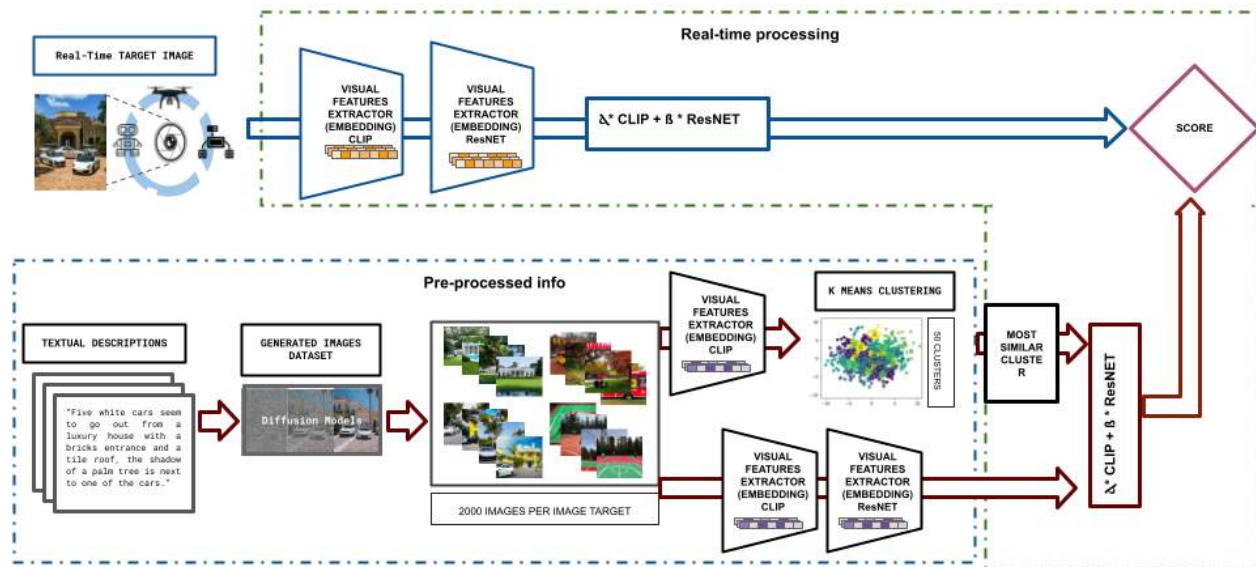


Fig. 1. We propose a methodology divided into two main stages: one for real-time processing and another for data preprocessing. The former observes the environment through the drone camera, obtaining its visual embeddings from CLIP and its visual descriptors from ResNet, which are then combined into a new proposed score. It is compared against the score obtained from the preprocessing stage, where the model previously received textual descriptions of the target destination and generated a pool of images through *Stable Diffusion*. These images are then reorganized into clusters, and the most similar cluster is retrieved to compare all visual embeddings and visual descriptors of all images within the cluster against the drone camera view

Building upon this study and motivated by drone delivery tasks that require fast decisions driven by factors such as battery life, operational efficiency, and mission effectiveness, which are crucial, we present a methodology for a real-time system capable of providing immediate responses to what the drone is observing (through its onboard camera) regarding the recognition of unseen destinations through generated images from textual descriptions.

Our methodology consists of two main stages: i) Data generation and organization, and ii) Real-time querying. In the first stage, we use textual descriptions to generate images offline using *Stable Diffusion* [24] to “imagine” the target location. These images comprise a dataset of 10,000 samples, which were organized via a

clustering technique to seek out the best image or images that semantically and visually match the camera images (query images) captured on board the drone. For the matching similarity, we explore the combination of two scores obtained when comparing CLIP embeddings and ResNet feature vectors computed for the query image and the images in the generated datasets. This enables the system to continuously retrieve the best-matched images, with a score indicating whether or not the current location is the target destination, a suitable capability for real-time performance.

The study in [19] presented five specific outdoor images referred to as *target images* used as control images to assess the effectiveness of generating images from text. For this work, we used the same images as examples of outdoor

target destinations, similarly serving as control images to evaluate our proposal. Thus, we placed these *target images* in random locations and conducted five flight sequences. Each sequence followed the same trajectory, but the *target images* were reordered each time. The *target images* were observed by the drone, covering the entire field of view of the camera, simulating real locations along the rest of the trajectory.

Our experiments aim to determine whether or not the searched target location received the highest score throughout the trajectory. We achieved a response with a frame rate average of 18 Hz. This frame rate is sufficient for online image processing during flight operations, a crucial aspect that enables deploying the drone in real-life applications.

This paper has been organized as follows: the related work is presented in Section 2; our methodology is described in Section 3; the experimental framework is presented in Section 4, and finally, our conclusions are drawn in Section 5.

2 Related Work

In recent years, Deep Learning (DL) and Artificial Intelligence (AI) have rapidly improved, offering the possibility of being used in various applications. Recently, considerable efforts have been directed towards the development of multimodal approaches. In particular, some methods merge Computer Vision and Natural Language Processing, making it possible to handle highly complex tasks including Image Captioning, Visual Question Answering, Image-Text Retrieval, Object Detection with Natural Language Description, and Text-To-Image (TTI) generation models, among others [28].

TTI generation models involve the use of neural networks to generate new images that either resemble a specified set of input images or create diverse variations, resulting in creative outputs [21, 13, 15]. One of these is the Variational Autoencoders (VAEs), a generative learning model that combines a likelihood distribution with neural networks [16, 23]. By mapping input data to a latent space following a Gaussian distribution, VAEs can generate synthetic data that closely

mimics the input dataset while predicting the variance and mean of the distribution [3, 6].

Generative Adversarial Networks (commonly known as GANs) have emerged as another significant TTI generative model producing images with improved fidelity [10]. GANs consist of a generator and a discriminator. The former attempts to create an image while minimizing the likelihood of it being classified as fake [2]. On the other hand, the discriminator receives the image and determines whether it is real or fake based on a binary decision. One of the key advantages of GANs is their ability to generate very similar data; however, they can lead to mode collapse, producing a limited variety of outputs.

Another recent approach to TTI generation involves Diffusion Models (DMs) [2, 14, 24]. Being a probabilistic model, such as VAEs [23], DMs learn a data distribution by denoising a normally distributed variable. Given an input image x , and by applying a bit of Gaussian noise and repeating this process N times, we obtain a sample of pure Gaussian noise. Subsequently, DMs learn to produce a denoised variant of the original image x , gradually removing the noise, with the expected output being a coherent image. DMs achieved impressive results, particularly in the realm of generative modeling and image synthesis, including tasks such as image generation, image segmentation [1], super-resolution [17, 25, 31], and TTI [2]. Also, DMs outperformed the state-of-the-art with [7] showing its potential in creative contexts, but also in other areas, such as robotics.

Once images are generated, it is desirable to obtain a numerical representation to compare them either against other generated images or real images. For instance, in image retrieval, where image search is involved, the bag-of-words method was the gold standard for image comparison [9]. In recent years, methods based on visual embeddings have become more effective in representing visual concepts that can be utilized for image comparison. In particular, there is CLIP (Contrastive Language-Image Pre-Training) [22], which is a neural network that learns visual concepts from natural language supervision, providing flexible predictions that enhance its

generalization capabilities. Unlike traditional models, CLIP recognizes a wide variety of visual concepts in images and associates them with their names using an image and a text encoder. The ultimate goal of CLIP is to align the image encoder embeddings with the text embeddings, thereby representing the information within the same multi-dimensional space, resulting in a 512-element embedding for each input.

Regarding image search, such data first has to be organized in order to achieve an efficient searching scheme. To this end, an option is to leverage the advantages of Unsupervised Machine Learning (UML) [11], thus learning from data without human supervision. UML aims to discover patterns and insights without any explicit guidance[20], it encompasses three main data organization types: *i. Association Rules:* This strategy aims to find associations or relationships between items in the dataset. *ii. Dimensionality Reduction:* By reducing the number of features or dimensions in the dataset, this strategy aims to make it easier to visualize the data. *iii. Clustering:* By exploring raw and unlabeled data, this method can break it down into groups based on similarities or differences. This type of analysis is useful when dealing with large input data that is believed to share some similarities but may not be readily separable. UML algorithms can render the input data more comprehensible and adaptable.

Finally, several studies have emerged where generative models have been used in different applications. For instance, in [26], GANs are used to imagine and fetch an object based on the user's speech command. The work in [27] uses CLIP and Diffusion Models to create scenes based on textual descriptions, synthesizing images to create visually augmented representations for each text. The work in [29] explores text-to-image synthesis, focusing on generating intricate visual content from abstract text prompts. A similar idea is explored in [15], which rearranges objects by inferring text descriptions and physically executing the rearrangement task, all without prior training data, also empowered by generative models. Following this trend of applications, we propose a methodology to employ DMs to 'imagine' destinations.

In contrast to other works, we aim to use DMs to generate useful information and recognize unseen destinations through visual embeddings, pushing for real-time processing.

3 Methodology

Our proposed research introduces an approach inspired by the way humans form mental images of places. Given a textual or spoken description of a location, the method aims to recognize an unseen target destination by leveraging Stable Diffusion to generate relevant visual representations (See Fig.1).

3.1 Dataset

Based on [5, 19], we depart from having five outdoor *target images* depicted in Fig. 2.

The images were sourced from the Internet and selected to serve as control images representing destinations the drone is intended to reach. The reason for using these *target images* instead of real ones taken from the environment is that Stable Diffusion was trained on landscapes and places similar to specific regions, so the generated images resemble the housing style in the USA. Therefore, Stable Diffusion might not visually resemble places shown in other regions, such as Mexico. Places, such as a basketball court, are more generic.

However, concepts like parks, luxury houses, and hot-dog carts can vary between Mexico and the USA. For convenience, the five *target images* were tagged as follows: (1) A house with cars parked in front of it (LUXURY); (2) A food truck selling hot dogs (HOT-DOG); (3) A kiosk (KIOSK); (4) A basketball court (BASKET); and (5) A house with a swimming pool (POOL).

We used the dataset presented in [5]. This dataset contains images generated from textual descriptions provided by three annotators. Each annotator provided ten textual descriptions for each *target image* that were fed to Stable Diffusion for generating a pool of 5,000 images. For this work, we tuned the model to generate 10,000 images in total, 2,000 images for each target, seeking to augment the visual variability of the generated images.



Fig. 2. Target outdoor images taken from [5]

To this end, we used the three best textual descriptions for each *target image* according to the assessments carried out in [5].

3.2 Image Feature Extraction

Visual embeddings are a dense representation of image features. There are different options to generate an embedding from an image. Currently, two of the most popular methods are the CLIP [22] and VisualBERT (VB) [18]. Assuming we have an embedding for the *target image* e_t and one for the image generated e_s , the similarity score can be calculated using the cosine distance between these two embeddings as follows:

$$score_C = 100.0 \times \cos(e_t, e_s) = 100.0 \times \frac{e_t \cdot e_s}{\|e_t\| \times \|e_s\|}. \quad (1)$$

We obtained a visual embedding for all images in the dataset and then compared it against each *target image* through the cosine similarity score. The resulting CLIP and VB scores were then sorted to identify which images had the highest scores.

3.3 Data Grouping

In a practical scenario, once the drone has navigated to the vicinity of the target destination, we want it to observe its surroundings and, based on the provided textual description, identify the most similar and, therefore, correct target destination. However, searching for the generated

image with the highest score in the whole pool of generated images is a time-consuming task. If we aim to enable a flying drone to match its camera view against the generated images visually, we require a strategy that reduces the search space.

For the above, we propose to organize the dataset using the K-Means clustering algorithm [29]. We aim to cluster the generated images such that the centroid of each cluster becomes the prototype indicative of the images within it. We tuned this clustering method to produce 50 clusters. We assessed the usefulness of the embeddings obtained by CLIP and VB to generate the clusters. We applied the Elbow method to determine the optimal K value for k-means clustering, using the 2,000 VisualBERT and CLIP visual embeddings as input for the Elbow method.

In both cases and for all target images, we found that 50 was the average optimal value. Once the clusters were formed, the embedding of the target image was compared against all the centroids using the cosine similarity. The generated image with the best score is retrieved from the cluster whose centroid achieves the highest similarity score to the target image.

3.4 Visual Comparison with ResNet

We hypothesize that the cluster with the highest similarity score is likely to contain the generated image that best matches the *target image*. However, the generated image can match the *target image* semantically but not visually.

For instance, consider the images shown in Fig. 3. The first image corresponds to one of our *target images* (POOL). The other three images are part of our dataset generated with Stable Diffusion. Image 1 (the target) and Image 2 are very similar visually, but Image 2 lacks a pool, a key element of our destination, resulting in a score of 0.79. However, the first and third images are semantically similar but not visually, resulting in a score of 0.85.

The first and last images are both semantically and visually similar, with neither lacking a pool or house and even having the same perspective, yet they return a score of 0.77. This presents a clear problem: it is insufficient to use the visual embedding and its similarity score to find the best



Fig. 3. The first image belongs to the target images (POOL). Images 2 to 4 were generated through Stable Diffusion and are part of our dataset of generated images

match between the target and generated images, as the best score came from the image that only resembled the target semantically.

To address the above, we propose using ResNet-18 [12] as a feature extractor, similar to a visual descriptor. ResNet-18 is renowned for extracting a feature vector of 512 visual characteristics containing visual features at high, medium, and low levels, and it has been widely used for image classification. Additionally, due to its robust feature extraction capabilities, it has demonstrated success in various computer vision tasks, such as object detection and image segmentation [30].

Since the cosine distance is less strict when evaluating two vectors, which would rank best if such vectors have a directional similarity, we propose to evaluate the visual similarity between two ResNet feature vectors $\mathbf{p}$ and $\mathbf{q}$ (where $\mathbf{p} = (p_1, p_2, \dots, p_n)$, and $\mathbf{q} = (q_1, q_2, \dots, q_n)$ are n -dimensional vectors), also using the cosine similarity defined in Eq. 1, but now as follows:

$$\text{score}_{\text{CR}} = 100.0 \times \cos(\mathbf{p}, \mathbf{q}) = 100.0 \times \frac{\mathbf{p} \cdot \mathbf{q}}{\|\mathbf{p}\| \times \|\mathbf{q}\|}. \quad (2)$$

Additionally, we utilize the Euclidean distance, defined as:

$$\mathbf{d} = \text{dist}_E(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^n (q_i - p_i)^2}. \quad (3)$$

Unlike cosine similarity, where larger values indicate greater similarity, Euclidean distance is a dissimilarity measure. The more similar the vectors, the closer the distance is to 0. Conversely, the Euclidean distance can approach infinity as

the difference between two vectors increases. Therefore, we normalised this distance using the distribution of our generated dataset, using the control images, as explained in Section 3. To this end, the mean Euclidean distance $\bar{\mathbf{d}}$ and the variance σ^2 were calculated among all generated images and the target image.

Therefore, assuming we calculate the Euclidean distance $\mathbf{d}$ between two ResNet feature vectors, the normalised score is computed as follows:

$$\text{score}_R = 1 - \frac{(\mathbf{d} - \bar{\mathbf{d}})^2}{3\sigma^2}. \quad (4)$$

Note that we divided by three times the variance σ^2 , where the majority of the scores fall within the distribution.

3.5 Linear Combination of Scores

Due to the CLIP similarity score potentially leading to the selection of images based solely on semantic similarity and that we also seek visual resemblance, we propose a linear combination to calculate an additional similarity score using the two scores for ResNet defined in Eq.2 and Eq.4.

Thus, we propose a linear combination weighted by α and β coefficients, first using the cosine similarity for both the CLIP embeddings and the ResNet feature vectors:

$$\text{score}_{\text{cosine-resnet}} = \alpha \cdot \text{score}_C + \beta \cdot \text{score}_{\text{CR}}. \quad (5)$$

Then, using the cosine similarity for the CLIP embeddings and the Euclidean distance for the ResNet feature vectors:

$$\text{score}_{\text{euclidean-resnet}} = \alpha \cdot \text{score}_C + \beta \cdot \text{score}_R, \quad (6)$$

where α takes the values of 0.0, 0.25, 0.5, 0.75, and 1.0, while $\beta = 1 - \alpha$, score_C and score_R represent the CLIP and ResNet scores, respectively.



Fig. 4. Top-view perspective image depicting the drone's trajectory path spanning 174.43 meters



Fig. 5. Representative image of the DJI Mini 3 Pro drone used to carry out the real-time experiments

4 Experimental Framework

Motivated by a delivery scenario and our previous works, we designed an experimental framework based on real-time response. Five test flights were conducted; in each flight, the same trajectory depicted in Figure 4 was followed. The *target images* were placed along the trajectory for the drone to observe, covering the entire field of view of the camera, and simulating real locations. However, for each test flight, the locations of the target images over the trajectory were rearranged.

Our first experiment aimed to recognize the target images as the target locations using the CLIP score solely. Following the outcomes of this experiment, we proceeded with two additional experiments using our new score based on the linear combination described in Eq.5 and Eq. 6. For these last two experiments, we present the results of our proposed score obtained from the linear combination of the CLIP and ResNet scores, using cosine distance for CLIP and both cosine and Euclidean distances for the ResNet score, respectively.

4.1 Hardware Equipment

For the following experiments, we used a DJI Mini 3 Pro drone to observe the environment (See Fig. 5). The generative and multimodal models were executed on a computer with an NVIDIA GeForce RTX 3050 graphics card with 4GB of VRAM, CUDA

version 12.2 with the 535.183.01 driver, and its respective torchvision library.

We also used this computer as a Ground Control Station (GCS) to communicate with the Drone for camera image acquisition. For the GCS, the ROS Noetic framework was run over Ubuntu 20.04 LTS.

4.2 Recognition Using Only CLIP

For our first experiment, we used the CLIP score to assess the similarity between the images observed by the drone's camera and the generated images. This experiment aimed to determine if a multimodal model like CLIP (to obtain a visual embedding) and its similarity score are sufficient to recognize previously unvisited locations (the target images). Fig. 6 shows the cluster names at the bottom of the graphs (LUXURY, HOT-DOG, KIOSK, BASKET, and POOL CLUSTER), indicating the target we are searching for.

The experiment is successful based on the highest score obtained throughout the trajectory. Suppose the graph's corresponding name, such as 'LUXURY CLUSTER' and its highlighted area (blue for LUXURY) show the highest peak over the entire trajectory, and there are no other peaks with the same value. In that case, we deem the experiment successful in recognizing the target destination. We depicted the scores obtained only in the first three flights, as the first drone flight yielded the best results, and the rest of the flight tests had similar results.

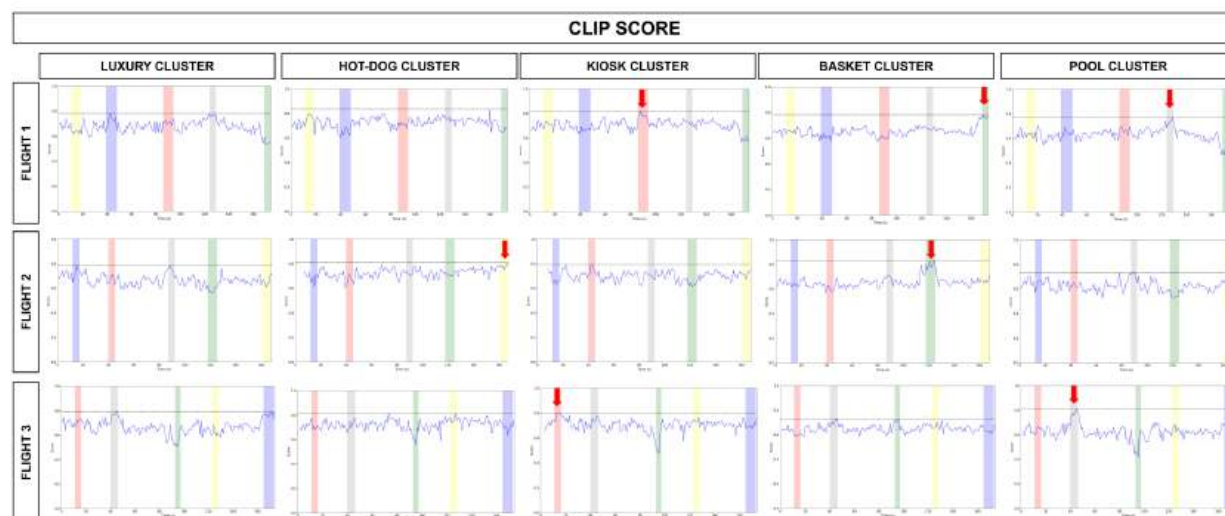


Fig. 6. The image depicts the results of drone flight tests one to three. Each image shows the CLIP score calculated using cosine distance plotted against time in seconds. Shaded areas represent the seconds during which the drone observed the corresponding target. The horizontal dotted line was plotted as a tangent to the highest scores obtained during the trajectory. The red arrow indicates the instances where successful recognition of the target was achieved

This experiment resulted in 18 out of 25 successful target recognitions, equivalent to an accuracy of 0.72.

4.3 Recognition with Semantic and Visual Appearance

We proposed to explore other similarity measures based on visual appearance, for instance, using feature vectors obtained from ResNet. However, conducting an experiment similar to the previous one, using only the ResNet score, was deemed impractical. ResNet focuses exclusively on visual similarities. Consequently, the generated images and the drone camera observations have a low probability of resembling each other, thereby eliminating the use of semantic information.

Given that the system continuously retrieves the cluster with the most semantically similar image within it, we can assert that the images up to this point meet this requirement but are not necessarily visually similar. Therefore, we proposed using ResNet as a visual feature extractor in a linear combination to add extra weight to the visual characteristics, ensuring that both criteria (semantic and visual similarity) are met.

We conducted two additional experiments using our proposed scores defined in Eq.5 and Eq. 6. These experiments involved varying the values of α and β in a range between 0 and 1 for every drone flight sequence. The first experiment using the proposed linear combination involved utilising the ResNet score calculated with Euclidean distance. Our findings indicate that the optimal values for α and β are 0.75 and 0.25, respectively. The system reached 19 out of 25 successful target recognitions, equivalent to an accuracy of 0.76. A summarized version of all linear combinations tested for this experiment using the flight test 1 sequence is depicted in Fig.7.

Similarly, the second experiment utilized the ResNet score, this time calculated with Cosine distance. Based on our findings, we selected $\alpha = 0.75$ and $\beta = 0.25$ as optimal values, aiming to achieve better results for our system by prioritising semantic characteristics over visual ones. Fig. 8 summarizes drone flight one with all different values of alpha and beta for this experiment.

Figures 9 to 13 summarize the results of the best linear combination. Our findings indicate that our newly proposed score is more accurate

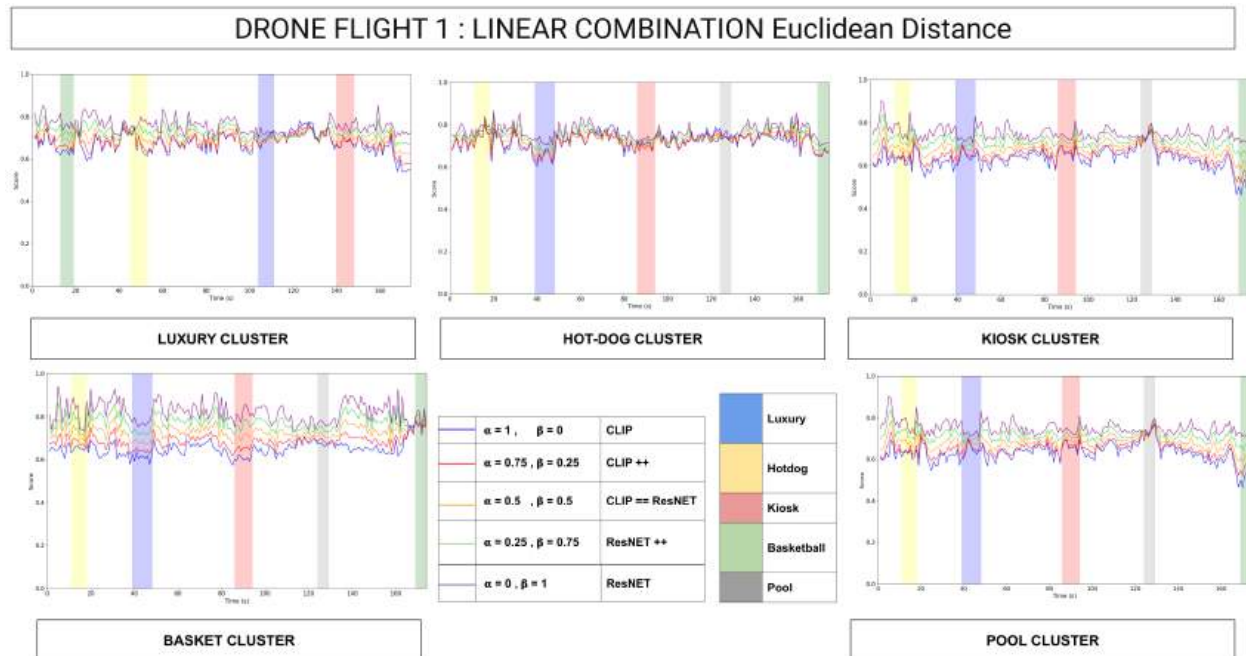


Fig. 7. The image displays all linear combinations tested on flight 1 trajectory with ResNet score obtained using Euclidean distance and by changing the values of α and β . The combination $\alpha = 0.75$ and $\beta = 0.25$ was selected as it shows the best results for coincidence between the target location and the highest score

than using CLIP or ResNet scores by themselves. CLIP gives more weight to semantic characteristics over visual ones, while ResNet only considers visual similarities; our proposed score combines both. For drone flights one and two (Fig.9 and Fig. 10), our proposed score effectively recognizes the target locations as the highest score over the entire trajectory.

However, it struggles when the system searches for the LUXURY HOUSE (Fig. 11, 12, 13) and has to differentiate between the LUXURY or the POOL target image, as the POOL target image can also be described as a “luxury house” due to its similar housing style. Conversely, when searching for the POOL target, the system effectively differentiates between LUXURY and POOL targets, due to the presence of the pool. The HOT-DOG target recognition also encounters difficulties in finding the correct destination.

In these cases, the system retrieves generated images that largely show a lot of trees. This problem is related to Stable Diffusion, which

does not always include all objects in the textual description in the generated image. Given that the environment used for the experiment also has views with many trees but lacks the hot-dog cart, the system cannot differentiate effectively. For the recognition of the KIOSK target, the system only fails once on flight five (Fig. 13), at the beginning of the trajectory, due to the drone observing a set of stairs with a railing, which could be mistaken for those of the KIOSK. Even when the system struggles in those cases, our proposed score combining the CLIP and ResNet scores yields better results in comparison with the rest of our experiments, resulting in 20 out of 25 successful target recognitions, equivalent to an accuracy of 0.80.

Finally, our system features an interface built in ROS, depicted in Fig. 14. Consisting of four windows: 1) Drone camera view: This window shows the drone’s camera view in real-time; 2) Grid of images: After capturing the environment, this frame is compared against the 50 centroids of

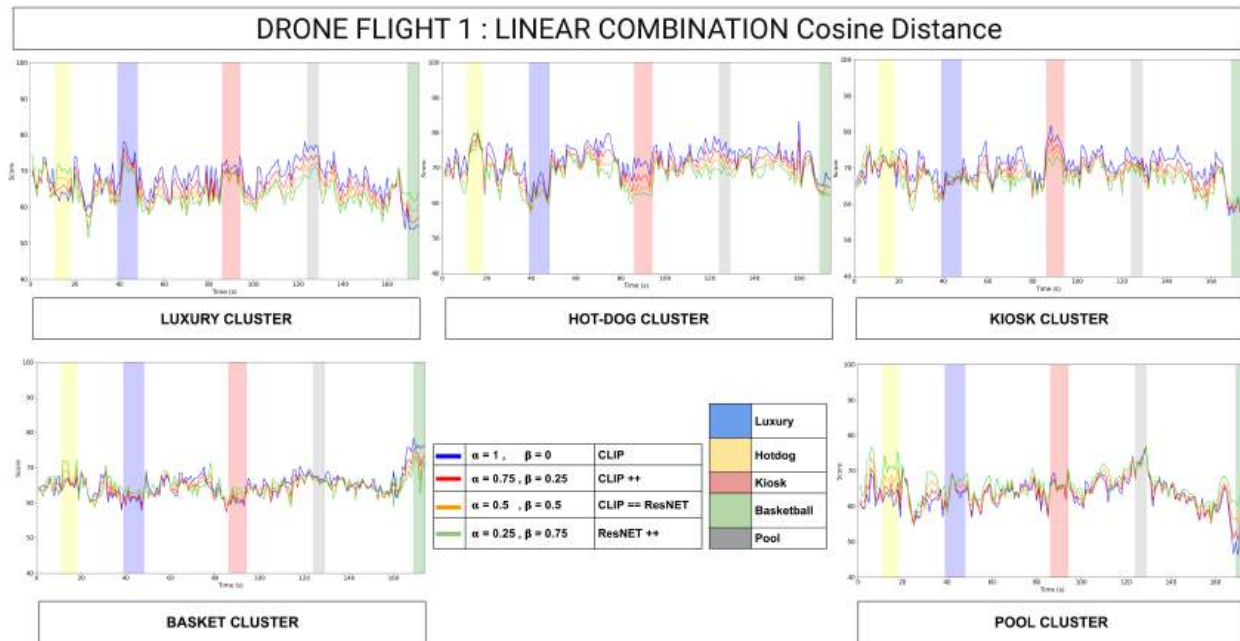


Fig. 8. The image displays all linear combinations tested on flight 1 trajectory with ResNet similarity obtained using Cosine Distance and by changing the values of α and β . The combination $\alpha = 0.75$ and $\beta = 0.25$ was selected as it shows the best results for coincidence between the target location and the highest score

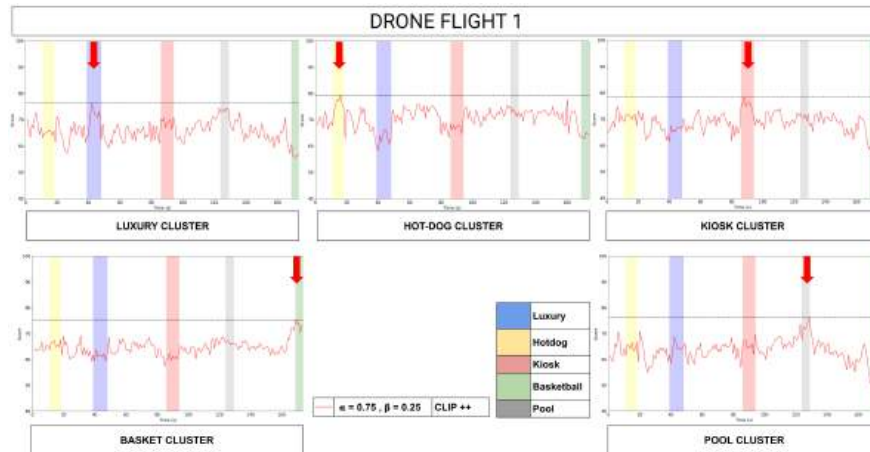


Fig. 9. First flight test: This figure shows the linear combination with cosine-ResNet scores over time. The highlighted areas indicate the seconds during which the drone observes the target image. Each plot represents the trajectory of the same flight path while searching for different target clusters. The red arrow indicates the instances where successful target recognition was achieved; all target destinations coincided with the highest score

our clusters to identify the best match; it displays the top 9 images within that cluster based on the CLIP score; 3) The three most similar images:

This window showcases the most similar image within the selected cluster, considering only the nine best images previously identified. Each image

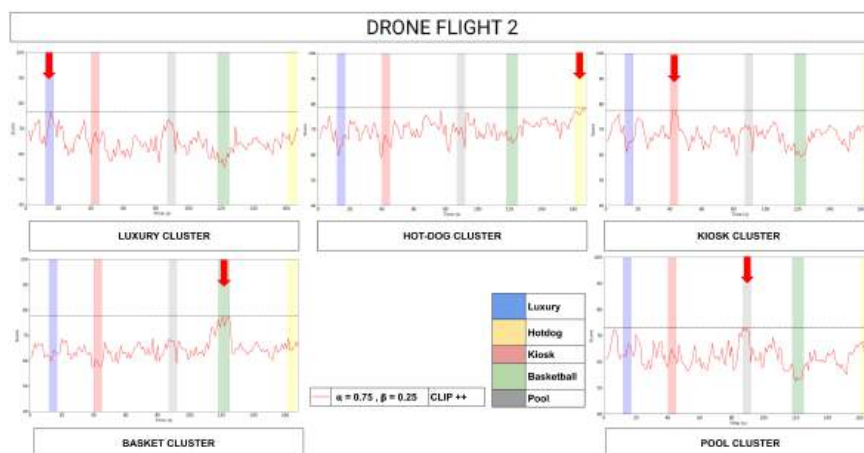


Fig. 10. Second flight test with cosine-ResNet: It effectively recognizes all target destinations as having the highest scores when the drone camera observes the target destination

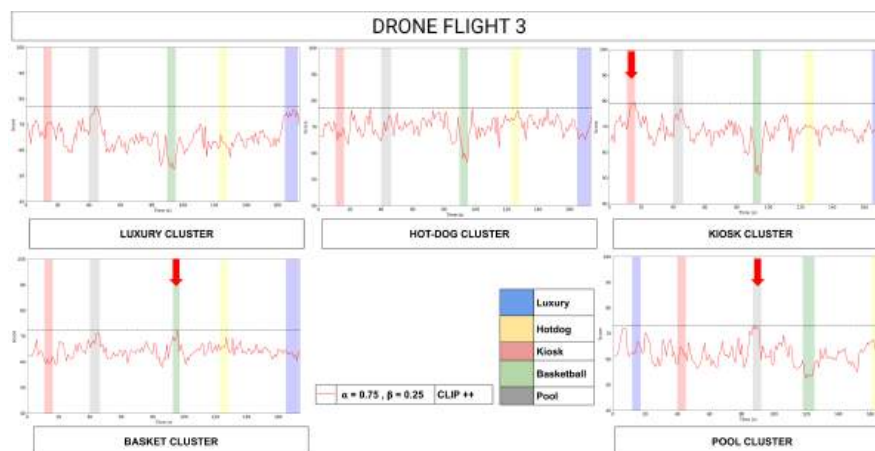


Fig. 11. Third flight test with cosine-ResNet: It struggles with the LUXURY CLUSTER to differentiate between Luxury and Pool targets, and also with the HOT-DOG CLUSTER, due to the cluster containing many images with trees, similar to the views observed in the drone's trajectory. On the other hand, it effectively recognizes the KIOSK, BASKET, and POOL targets

corresponds to one linear combination listed in the box, and the CLIP score is listed as the first row in the box only for comparison against the newly calculated scores. 4) Scores distribution: This window presents two graphs. From left to right, the first graph plots images against their scores, while the second graph illustrates a Gaussian distribution of all images in the retrieved cluster.

Higher scores in the first graph indicate a closer resemblance between the target location and the

drone's camera view. A video of the system can be seen here¹.

5 Conclusions

We have presented a methodology to address the real-time recognition of unseen destinations in the context of drone delivery. To achieve an efficient

¹<https://youtu.be/Pvt41hkSZIM>

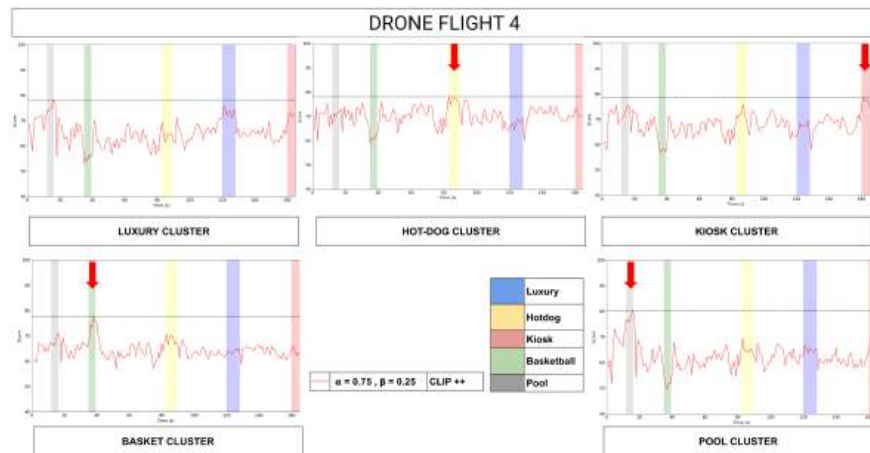


Fig. 12. Fourth flight test with cosine-ResNet: It effectively recognizes the HOT-DOG, KIOSK, BASKET, and POOL targets, but struggles to differentiate between LUXURY and POOL targets when comparing the target over the LUXURY CLUSTER of generated images

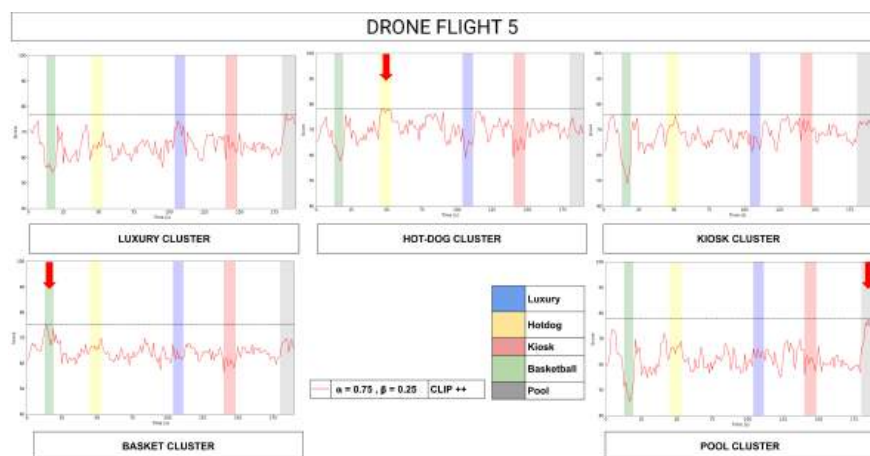


Fig. 13. Fifth flight test with cosine-ResNet: It effectively recognizes the HOT-DOG, BASKET, and POOL targets, but again struggles to differentiate between LUXURY and POOL targets when comparing the target over the LUXURY CLUSTER of generated images, and it also has difficulty recognizing the KIOSK target

performance, we have outlined a methodology using generative models to create a set of images that capture the visual and semantic features described by a textual description of a target location. We have proposed a clustering procedure to group these generated images into visually and semantically similar clusters, thereby reducing the search space.

Furthermore, we have investigated how to combine two scores derived from CLIP and

ResNet, achieving a recognition accuracy of 0.80. This computation is performed between images captured by the drone's onboard camera and key images ranked within the clusters, aiming to identify the most similar image.

Our system demonstrates the ability to identify unseen destinations solely from textual descriptions. Operating at an average rate of 18 Hz, our results show promising real-time response capabilities.

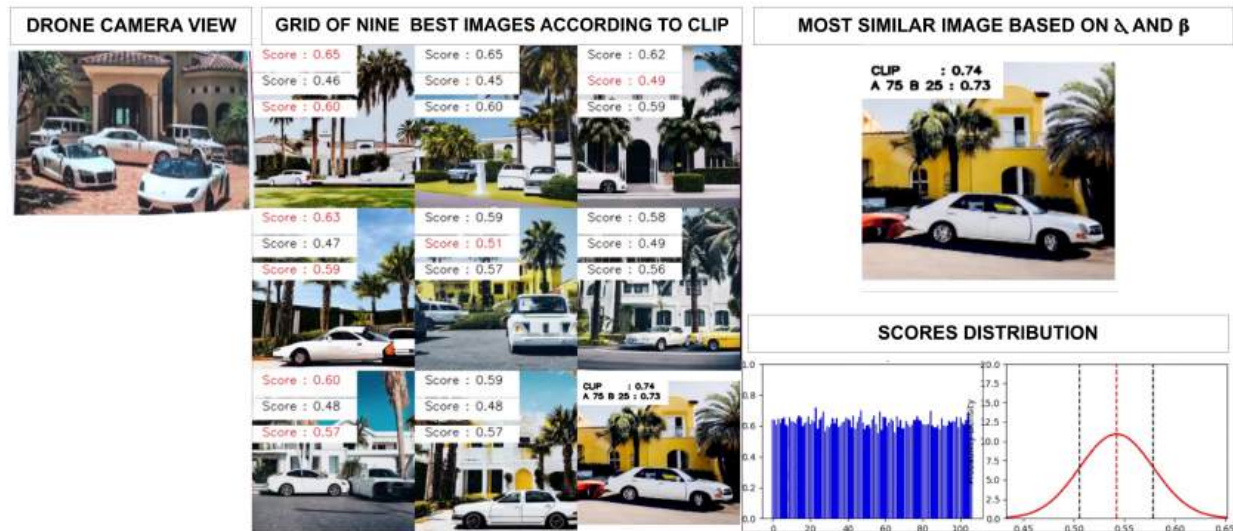


Fig. 14. System interface, showing the drone camera view and the best nine images within the most similar cluster according to the CLIP score. It also includes the most similar images for each linear combination depicted in the box and the score distribution of all images in the cluster

However, our work acknowledges limitations. Despite improvements using a linear combination with ResNet, our system still encounters cases where semantically similar destinations can achieve scores equal to or greater than the actual target.

Moreover, accurately generating images based on textual descriptions continues to pose challenges for our generative model.

A significant limitation of this work was the use of Stable Diffusion to generate target images. The model was primarily trained on images of locations and landscapes from the United States, which limited its ability to accurately represent local environments.

Adapting the model to local locations would have required additional training, involving substantial computational resources and time.

Consequently, for the field experiments, large printed versions of the target images were placed in outdoor environments and used as recognition targets. Although the experimental dataset was not exhaustive, the results demonstrate the potential of the proposed approach in terms of both computational efficiency and target recognition performance.

Further validation under a broader range of environmental conditions and target scenarios is planned for future work. We will also focus on improving our system's frame rate.

Additionally, we plan to explore other generative models that yield better results in image generation, emphasizing aspects such as temporal coherence and cardinality. Furthermore, we aim to investigate the use of video generative models, like SORA, to obtain temporal cues that could complement our approach.

Acknowledgments

The first author is thankful to the National Council of Science and Technology of Mexico (CONAH-CYT) for her scholarship number 1231026.

References

1. Amit, T., Shaharbany, T., Nachmani, E., Wolf, L. (2021). Segdiff: Image segmentation with diffusion probabilistic models. arXiv preprint arXiv:2112.00390.

2. **Bie, F., Yang, Y., Zhou, Z., Ghanem, A., Zhang, M., Yao, Z., ..., Song, S. L. (2023).** Renaissance: A survey into ai text-to-image generation in the era of large model. *arXiv preprint arXiv:2309.00810*.
3. **Blei, D. M., Jordan, M. I., Paisley, J. W. (2012).** Variational bayesian inference with stochastic search. *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pp. 1367–1374.
4. **Boysen, N., Fedtke, S., Schwerdfeger, S. (2021).** Last-mile delivery concepts: a survey from an operational research perspective. *Or Spectrum*, Vol. 43, No. 1, pp. 1–58.
5. **Carranza, J. M., Hernandez-Farias, D. I., Rojas-Perez, L. O., Ponce, A. A. C. (2023).** Why do i need to speak to my drone? 14th ANNUAL INTERNATIONAL MICRO AIR VEHICLE CONFERENCE AND COMPETITION, IMAV2023-7.3.
6. **Cho, K., van Merriënboer, B., Bahdanau, D., Bengio, Y. (2014).** On the properties of neural machine translation: Encoder–decoder approaches. *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, Association for Computational Linguistics, pp. 103.
7. **Dhariwal, P., Nichol, A. (2021).** Diffusion models beat gans on image synthesis. *NeurIPS*, Vol. 34, pp. 8780–8794.
8. **Dorling, K., Heinrichs, J., Messier, G. G., Magierowski, S. (2016).** Vehicle routing problems for drone delivery. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 47, No. 1, pp. 70–85.
9. **Escalante, H. J., Ponce-López, V., Escalera, S., Baró, X., Morales-Reyes, A., Martínez-Carranza, J. (2017).** Evolving weighting schemes for the bag of visual words. *Neural Computing and Applications*, Vol. 28, pp. 925–939.
10. **Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y. (2014).** Generative adversarial nets. *NeurIPS*, Vol. 27.
11. **Hahne, F., Huber, W., Gentleman, R., Falcon, S., Carey, V. J. (2008).** Unsupervised machine learning. *Bioconductor case studies*, pp. 137–157.
12. **He, K., Zhang, X., Ren, S., Sun, J. (2016).** Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.
13. **He, X., Deng, L. (2017).** Deep learning for image-to-text generation: A technical overview. *IEEE Signal Processing Magazine*, Vol. 34, pp. 109–116. DOI: 10.1109/MSP.2017.2709347.
14. **Ho, J., Jain, A., Abbeel, P. (2020).** Denoising diffusion probabilistic models. *NeurIPS*, Vol. 33, pp. 6840–6851.
15. **Kapelyukh, I., Vosylius, V., Johns, E. (2023).** Dall-e-bot: Introducing web-scale diffusion models to robotics. *IEEE Robotics and Automation Letters*, Vol. 8, pp. 3956–3963. DOI: 10.1109/LRA.2023.4541539.
16. **Kingma, D. P., Welling, M. (2013).** Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.
17. **Li, H., Yang, Y., Chang, M., Chen, S., Feng, H., Xu, Z., Li, Q., Chen, Y. (2022).** Srdiff: Single image super-resolution with diffusion probabilistic models. *Neurocomputing*, Vol. 479, pp. 47–59.
18. **Li, L. H., Yatskar, M., Yin, D., Hsieh, C. J., Chang, K. W. (2019).** Visualbert: A simple and performant baseline for vision and language. *arXiv*.
19. **Martínez-Carranza, J., Hernández-Farías, D. I., Vázquez-Meza, V. E., Rojas-Perez, L. O., Cabrera-Ponce, A. A. (2023).** A study on generative models for visual recognition of unknown scenes using a textual description. *Sensors*, Vol. 23, No. 21, pp. 8757.
20. **Nanni, L. (2006).** Cluster-based pattern discrimination: A novel technique for feature selection. *Pattern Recognition Letters*, Vol. 27, No. 6, pp. 515–519.

21. **Oussidi, A., Elhassouny, A. (2018).** Deep generative models: Survey. Proceedings of the 2018 International Conference on Intelligent Systems and Computer Vision (ISCV), Fez, Morocco, pp. 1–8.
22. **Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., ..., Sutskever, I. (2021).** Learning transferable visual models from natural language supervision. International Conference on Machine Learning, PMLR, pp. 8748–8763.
23. **Rezende, D. J., Mohamed, S., Wierstra, D. (2014).** Stochastic backpropagation and approximate inference in deep generative models. ICML, PMLR, pp. 1278–1286.
24. **Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B. (2022).** High-resolution image synthesis with latent diffusion models. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10684–10695.
25. **Saharia, C., Ho, J., Chan, W., Salimans, T., Fleet, D. J., Norouzi, M. (2022).** Image super-resolution via iterative refinement. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), Vol. 45, No. 4, pp. 4713–4726.
26. **Srichandan, A., Gupta, B., Prakasam, P. (2021).** Imagination with generative adversarial networks for object detection. 2021 Innovations in Power and Advanced Computing Technologies (i-PACT), IEEE, pp. 1–6.
27. **Tang, T., Chen, Y., Du, Y., Li, J., Zhao, W. X., Wen, J. R. (2023).** Learning to imagine: Visually-augmented natural language generation. arXiv preprint arXiv:2305.16944.
28. **Wang, L., Hu, W., Qiu, H., Shang, C., Zhao, T., Qiu, B., Ngan, K. N., Li, H. (2022).** A survey of vision and language related multi-modal task. CAAI Artificial Intelligence Research, Vol. 1, No. 2, pp. 111–136.
29. **Wu, J. (2012).** Advances in K-means clustering: A data mining thinking. Springer Science & Business Media.
30. **Zeiler, M. D., Fergus, R. (2014).** Visualizing and understanding convolutional neural networks. ECCV.
31. **Zhao, M., Bao, F., Li, C., Zhu, J. (2022).** Egsde: Unpaired image-to-image translation via energy-guided stochastic differential equations. NeurIPS, Vol. 35, pp. 3609–3623.

Article received on 11/06/2026; accepted on 10/09/2026.

** Corresponding author is Jose Martinez-Carranza.*