Generate Comic Strips Using AI

Perera M.P.M.[1], Gunasekara A.M.P.P.[1], C.D.Adhihetty[1], Kollure K.A.D.D.[1], Nuwan Kodagoda[1], Amitha Caldera[1]
[1] Faculty of Computing, Sri Lanka Institute of Information Technology
[1]New Kandy RD, Malabe, Sri Lanka
it20133818@my.sliit.lk, it20135270@my.sliit.lk, it20135966@my.sliit.lk, it20133672@my.sliit.lk, nuwan.k@sliit.lk, amitha.c@sliit.lk

***Abstract*— The art of storytelling through comic strips has long been a favourite, grabbing the interest of readers of all ages and developing into a variety of genres today, from superhero comics to political comics. Comic strip readers are more literate, empathic, and open to social issues, according to studies. Comics have the power to change the way we see the world. This research paper presents a system (ComicGenie) for automatically creating comic strips for Batman based on user-entered text descriptions. The proposed system consists of distinct components such as character detection, environment detection, text bubble generation, and voiceover over the scenario, each specializing in different aspects of comic strip creation. SVM and fastText technologies in NLP were utilized for text classification when developing models. Overall, this research contributes to the field of comic strip creation, offering a comprehensive web application. The ComicGenie holds significant potential to revolutionize the comic industry and inspire new avenues for computer-generated storytelling.**

Keywords*— *Comic strips, NLP, Text classification, FastText, SVM, ComicGenie

I. INTRODUCTION

Comic strips have a lengthy and varied history that dates back to the late 19th century [2]. The comics were first published in newspapers and magazines. The renowned comic strip "The Yellow Kid," produced by Richard F. Outcault in 1895, is regarded as one of the first in the comic genre [2]. Comic strips flourished in the twentieth century, becoming a significant part of popular culture.

The traditional method of making comic strips was drawing comic characters and environments by hand. Using pencil and paper, the characters came to life with unique personalities and stories. Each comic has a unique story and drawings and everything depends on the creator's imagination. Each stroke of the pencil added depth to the characters and the dialogue, making each comic strip a unique creation.

When digital stuff, the internet, and technologies improved, comic strips also improved day by day. Webcomics became a magical thing for everyone who likes to read and create comics. It's giving artists a fresh way to show off their work to the whole world [5]. Creators didn't have to deal with traditional ways of putting things out there anymore. This democratization of comics allows diverse writers to explore different storytelling styles and themes.

Crafting novel characters, settings, and dialogues that harmonize with the comic's overarching ambience and manner poses substantial hurdles [6]. Devoting undivided time and focus to originate and illustrate new comics might lead to neglect of vital responsibilities or facets of life. Generating original settings for comics demands additional time and exertion. Even though there are many digital comic tools available right now, comic book creators must use different devices for each task [4].

The resulting system introduces the revolutionary creation of comic strips with character images, background images, dialogues, and background sounds, a groundbreaking solution that ensures inclusivity [5] and accessibility for all, including those who are blind or have low vision. In a world where comics have captured the imagination of millions recognize the need to extend this pleasure to every individual, regardless of the visual capabilities.

Presently, no single tool exists that can seamlessly integrate all essential elements, such as characters, backgrounds, voices, and dialogues, for comic artists. However, this innovative system is here to bridge that gap. By harnessing the power of cutting-edge machine learning technologies, this platform allows users to input text and effortlessly create captivating comic strips, personalized to the details provided in the text input [3].

The primary objective behind this creation is to produce comic strips that go beyond mere visuals, crafting enthralling stories with engaging dialogue and compelling plotlines. Firmly believing that a comic strip can be both visually appealing and intellectually stimulating, it caters to the diverse tastes of comic enthusiasts. One of the key highlights of this system lies in its remarkable image-detection capabilities [1].

In conclusion, this creation strives to revolutionize the comic industry by providing a user-friendly and AI-driven platform that crafts comic strips with diverse characters, captivating backgrounds, engaging voices, and thought-provoking dialogues, all inspired by user-generated text inputs. Welcome to a new era of comics, where imagination and technology unite to bring joy to every individual, regardless of visual abilities.

II. LITERATURE REVIEW

ComicGen is a system that uses a Variational Autoencoder (VAE) to produce comic panels based on user inputs, as presented by Zhang et al [11]. To produce compelling and logical comic storylines, the technique combines user-generated textual descriptions [11]. The method creates a variety of character expressions and visual settings that are consistent with the narrative by incorporating contextual information into the VAE, demonstrating the potential of using user-generated text for comic creation. ComicChat is a system that Liang et al. suggest focuses on dialogue-driven comic creation [12]. Natural Language Processing (NLP) techniques are used to turn user-supplied text dialogues into comic panels. [12]. Here also incorporate sentiment analysis to make sure that characters react emotionally in line with expectations, which adds to the realistic nature of the humorous encounters.

Shibly et al. investigate the use of artistic style transfer in AI-generated comic strips [13]. The work entails converting the aesthetic of existing comics to AI-generated panels, producing visually coherent and stylistically consistent storylines. The authors show how AI-generated content can mix smoothly with known artistic styles by using convolutional neural networks (CNNs) and style loss algorithms [13]. Wu et al. explore personalized comic strip development by focusing on character-specific dialogue generation [14]. The technique makes use of recurrent neural networks (RNNs) to extract personality traits and preferences from user inputs [14]. The authors improve the authenticity and relatability of AI-generated humorous exchanges by ensuring that character dialogues match the personalities. Ingold et al. explore the incorporation of weather as a narrative element in AI-generated comic strips. Weather-inspired visual tales are introduced in the work, in which AI models produce scenes and character interactions based on weather data [10]. The authors broaden the narrative elements of AI-generated comics by incorporating meteorological data and climate-related emotions [7]. Park et al. investigate the use of weather as a narrative element in AI-generated comic strips [15]. Weather-inspired visual tales are introduced in the work, in which AI models produce scenes and character interactions based on weather data [15]. The authors broaden the narrative elements of AI-generated comics by incorporating meteorological data and climate-related emotions [9]. Smith and colleagues investigate the ethical elements of AI-generated creative work, such as comic strips [8]. This talks about worries about content ownership, copyright difficulties, and the possibility of bias in AI-generated tales [8]. The significance of developing ethical norms and frameworks to handle the legal and societal ramifications of AI-generated material in the creative realm is emphasized in the article.

The proposed system consists of distinct components such as character detection, environment detection, text bubble generation, and voiceover over the scenario, each specializing in different aspects of comic strip creation.

III. METHODOLOGY

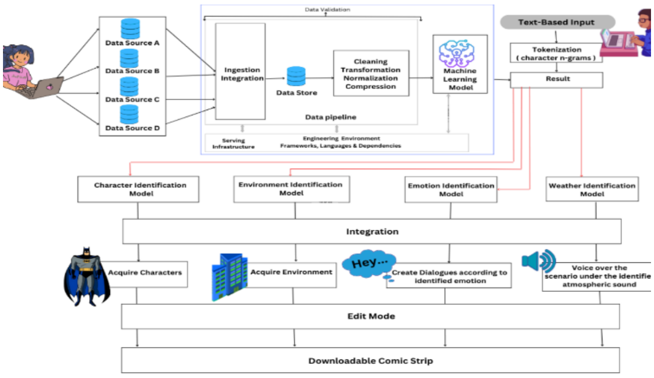


Fig. 1. Overall system diagram of the proposed solution

The system overview diagram presents the complicated data-driven process. The first stage involves creating separate datasets for characters, environments, emotions, and weather. Data was acquired in collaboration with artists and illustrators to compile images and used a variety of sources including Kaggle. Following that the datasets undergo rigorous data validation, which includes cleaning, transformations, normalization, and compression. Machine learning models help with this key stage.

The frontend, built on React, communicates with the backend in real-time, allowing users to see predicted character and environment comic images. An API generates dialogues based on the predicted emotion. Similarly, the detected weather type causes background music to be played in conjunction with the generated voices via another API. The final phases allow readers to download the finished comic strip in both PDF and MP4 formats.

A. Build Characters

With the introduction of "ComicGenie," creating comics has become quick and simple in a diversified environment bursting with myriad ideas. With the help of user-provided descriptions, this ground-breaking approach will let illustrators produce fascinating comic characters. By coordinating user-provided character descriptions with appropriate character photos from its large dataset, ComicGenie accelerates the character selection process by utilizing the strength of Support Vector Machines (SVMs).

1. *Data Gathering*:

The core of ComicGenie is its meticulously selected dataset, which includes a wide range of characters from different genres and artistic movements. This dataset was carefully put together to represent the essence of many character archetypes, giving users a wide range of alternatives. To construct a complete collection of character images that cover a wide range of features, appearances, and personalities, gathering this data requires intensive research, curation, and collaboration with artists and illustrators.

2. *Data Preprocessing:*

Before using SVMs, data preparation is critical for boosting the accuracy and efficiency of the ComicGenie system. Several preprocessing rules are applied to the unprocessed character descriptions and related images. A solid representation is created by structuring textual descriptions and extracting the pertinent elements. To extract relevant character properties, one may use text analysis methods like natural language processing (NLP), which includes tokenization, stemming, and deleting stop words.

3. *Support Vector Machines (SVMs):*

To classify characters, ComicGenie uses Vector Machines, a key component of its functionality. SVMs are supervised learning algorithms created primarily for regression and classification tasks [18]. SVMs are trained using the pre-processed data in the context of ComicGenie, with relevant character categories serving as labels and text descriptions acting as input features. By recognising patterns and relationships in the data, the SVM algorithm creates decision limits that efficiently categorize character qualities based on user descriptions [22].

4. *Model Training and Tuning:*

SVMs require a rigorous training process before predicting characters with accuracy. To enable the SVM model to iteratively learn and fine-tune its parameters. The dataset is divided into two parts: training and validation. Hyperparameter adjustment is also performed to increase the model's performance and achieve a balance between specificity and generality. Through this procedure, ComicGenie is guaranteed to accurately translate and pair user-provided character descriptions with relevant character images.

5. *Character-Image Mapping:*

The best appropriate character images are automatically mapped to user descriptions by ComicGenie after the SVM model has been properly trained and optimized. When a user inputs a description, the SVM model evaluates the data and forecasts the character archetypes or traits that are most relevant. The system then finds the equivalent image in its vast collection, ensuring a humorous character that is attractive and interesting to the user [21].

B. Build Environment

Comics are a great storytelling technique because the cartoon-like format enables the realistic portrayal of feelings, environments, actions, and innovative worlds. Background pictures in comic strips are essential to the medium's overall storyline and visual appeal. The reader's understanding and sense of immersion in the story are greatly enhanced by these backdrops, which provide the context for the character's actions and dialogue. Based on user-provided descriptions, this "ComicGenie" tool is designed to assist artists in creating compelling comic backgrounds.

1. Data Collection and Processing

The comic background image dataset includes a wide range of visual settings and situations, each of which is identified by a name and a description. These selected backgrounds create engaging backdrops for comic strips and visual narratives. To develop a strong association between the background image names and the associated descriptions, a detailed examination of each image name was carried out. By considering linguistic messages, context, and visual components displayed in the images, the goal was to determine the fundamental atmosphere, mood, and qualities indicated by the names.

2. Fasttext

FastText is frequently used for tasks like sentiment analysis, text categorization, and more. The Facebook AI Research (FAIR) team developed a library and toolkit for effective text classification and word representation. It was created to handle text data effectively and efficiently, especially in situations where there is a lot of text and there aren't many computational resources available [24]. Word embeddings, which are vector representations of words in a continuous space, can be created using FastText [16]. By capturing the semantic relationships between words, these embeddings make it possible for machines to understand the context and meaning of words more accurately.

3. Model Training

The dataset includes training and validation sets. Use the training set to train the FastText text categorization model. It contains a piece of the pre-processed data, which includes names for the background images as well as textual descriptions for those images. A text classification model that links textual descriptions to backdrop image names is trained using FastText. The resulting model can anticipate background names from fresh descriptions, making it easier for comic strips and graphic storytelling to seamlessly combine text and graphics [24].

4. Environment-Image Mapping

After the model has been appropriately trained and optimized, ComicGenie automatically maps the environment photos to user descriptions. The model analyses the data once a user enters a description and predicts the environment archetypes or features that are most important. The system then retrieves the relevant image from its vast collection to create a visually pleasing and engaging environment that is in keeping with the user's vision.

In conclusion, ComicGenie makes a significant addition to the art of creating comic strips by providing a quick and easy approach to captivating backdrop images based on user text inputs. This strategy holds promise for engaging fans with compelling and visually appealing comic strips by transforming the process of comic creation.

C. Build Dialogues

Creating dialogues for comic strips is an important part of bringing characters to life and giving dimension to the interactions. Emotional dialogue is critical to reader engagement, empathy, and creating a deeper connection between readers and the comic universe. To accomplish this goal, a machine learning (ML) model is proposed to automatically identify the emotions within a given description. An API generates dialogues that are smoothly aligned with the emotional context based on effectively recognised emotions.

1. Data collection and Preprocessing:

An emotion detection model is trained using a curated dataset of descriptions that have been annotated with the associated emotions. Gathered a wide range of data from reliable sources, including Kaggle, and expanded it to make sure that emotions were fully represented. Here, specifically on four key emotion categories were focused on in particular: funny, angry, thriller, and neutral, as these emotions cover a wide range of attitudes frequently seen in comic strips.

2. NLP-FastText

For the challenge of emotion recognition, various ML algorithms were studied, each differing with benefits, restrictions, and amount of accuracy. The FastText library stood out among NLP algorithms due to its competence in handling text classification tasks, showing higher accuracy values, and its ability to provide robust results. FastText, an extension of the Word2Vec algorithm, represents words as character n-grams, which makes it possible for the model to effectively capture sub-word information [17]. This feature is especially useful when dealing with emotional statements contained in user-entered descriptions.

3. Model Training:

The processed dataset was divided into training and testing sets before being used to build the emotion detection model. The model learns to associate the description's content with the appropriate emotion labels throughout the training phase. The trained model is then utilized to predict emotions based on new text inputs [19]. This prediction skill highlights the model's ability to extract emotions from text descriptions. The 'train_test_split' function is essential for distinguishing between data used for model training and data needed to test its generalisation.

4. API Integration:

When the emotion recognition model successfully recognises emotions inside user-entered descriptions, an external API is invoked to generate dialogue consistent with the identified emotional context. This API integrates smoothly into the research pipeline, allowing for the construction of conversations tailored to specific emotion kinds [23].

D. Voice Over the Scenario

Comic book enthusiasts are in for a unique encounter in a world teeming with unique thoughts and artistic concepts. A cutting-edge feature that brings narratives to life enables the incorporation of many voices and background weather noises into comic storytelling. With the help of this ground-breaking feature, users may lose themselves in a multisensory comic adventure where weather forecasts, recognisable voices, and evocative noises all work together to create an unparalleled storytelling experience.

1. *Weather-Driven Audio Atmosphere:*

A complex audio system that responds to the user's provided weather description is at the core of this experience. A specifically trained FastText model gets to work as users enter weather-related information into the description text field. The input is processed by FastText, a small text representation package, which determines the predominant weather scenario, such as a sunny day, a rainy afternoon, a snowy landscape, or any other possible atmospheric situation.

2. *Training the FastText Model*:

The FastText model is built by thorough training on a carefully selected dataset. This collection includes a wide range of descriptions of the weather together with matching audio profiles. The algorithm improves its capacity to precisely forecast the most appropriate audio ambience for a given input by learning to correlate distinct textual clues with particular weather patterns [20].

3. *Dynamic Weather Sound Selection:*

The feature automatically chooses an audio background that complements the comic's environment based on the weather condition that was detected. There is a ready supply of precisely recorded weather sounds, from light rain to screaming winds. Users are immersed in an immersive audio world that supports the visual narrative thanks to this dynamic selection procedure.

4. *Voice Modulation and Third-Party API Integration:*

A third-party API takes the stage, adding a further element of interest by offering a variety of distinctive tones and voices to narrate the comic plot. The story develops line by line, and each passage is improved by a vocal identity that matches the emotions and characteristics of the characters. The use of various voices gives the comic's dialogue and monologues life and elevates the experience to the level of an audio masterpiece.

5. *User Experience Unleashed:*

After the scene is set, users are brought into a world where the comic's images, weather noises, and narrative voices smoothly blend. The auditory landscape dynamically changes as moving through each frame, enhancing the impact of the story's climactic moments and encouraging a strong emotional connection.

In conclusion, a new era of immersion and engagement is introduced by the incorporation of dynamic audio aspects into comic storytelling. Users are given a comprehensive sensory experience that blends fantasy and reality by utilizing FastText's weather recognition, third-party APIs for various vocal tones, and a collection of carefully collected weather sounds. This combination of music that is influenced by the weather and comical graphics takes the narrative to new heights and invites audiences to go on a multi-dimensional journey that awakens the senses and moves the soul.

IV. RESULTS AND DISCUSSION

A variety of machine learning techniques were employed to train the models aimed at addressing Character identification, Environmental identification, Emotional identification, and Weather Identification. The objective of this procedure was to ascertain the architecture capable of yielding the most optimal performance for each distinct problem. The effectiveness of these architectures was gauged through an assessment of the performance of accuracy. This involved the measurement of outcome predictions by the models after each iteration of the optimization process. The architecture selected for each problem was that which showcased the highest accuracy among all the architectures under consideration. This approach enabled the selection of the architecture that displayed the most optimal alignment between predictions and actual outcomes.

For these models, a meticulously curated dataset is found, encompassing a diverse array of characters, environments, emotions, and weather conditions across a spectrum of genres and styles.

After the models were trained, the selection of the optimal architectural models was driven by the attainment of the highest testing accuracies. A testing accuracy of 85% was achieved by the character identification model, which was implemented using the SVM algorithm. Furthermore, the environment identification model demonstrated a testing accuracy of 84% through the utilization of the FastText technique. In the case of emotion identification, preference was given to the model with the highest testing accuracy, which reached 87%. Similarly, for the identification of weather conditions, the FastText architecture was chosen as the optimal solution, showcasing a testing accuracy of 88%.

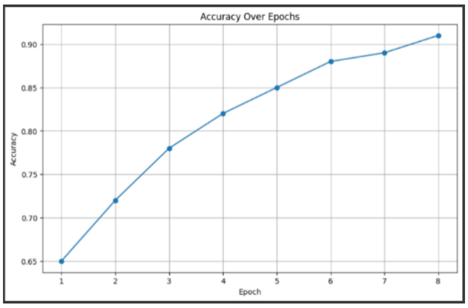


Fig. 2. Accuracy graph for the weather identification

This graph displays the accuracy scores obtained during the model training iterations. This can help to see how the accuracy changes over time.

Figure 3 illustrates the confusion matrices related to Character emotion identification which help to evaluate the performance of the trained model.

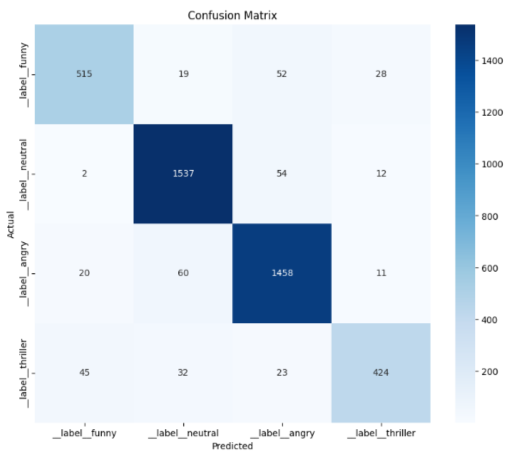


Fig. 3. Confusion matrix for the emotion detection model

When faced with the task of classifying a complete text, such as a sentence or document, FastText develops extensive knowledge by calculating a simple average of the text's word vectors. This combination yields the "text representation vector," which serves as a condensed representation of the overall meaning of the text. FastText attempts to distil the substance of the text into a manageable vector by examining the contributions of individual words.

FastText extends this concept to classes. For each class in the classification issue, there is a "class representation vector." These vectors represent the distinct characteristics and patterns associated with each class. This dynamic vector utilization serves as the foundation for the upcoming categorization step.

The classification probability is computed using the softmax function, which computes the dot product of the text representation vector and the class representation vector. This computation yields a probability distribution across all possible classes.

Mathematically,

$$P(class \mid T) = softmax(v(T) \cdot v(class)) \quad (1)$$

where,

v(T): Text representation vector.

v(class): Class representation vector.

P (class | T): Probability of the text belonging to the class.

FastText's categorization accuracy is greatly improved by the training procedure. From labelled training data, the model learns both the class representation vectors and the word embeddings. These vectors are optimized through iterative changes to minimize classification loss. This includes improving the vectors so that they forecast high probabilities for the correct classes while reducing those for the erroneous ones.

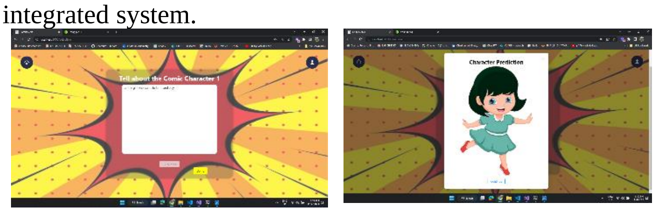
Because of its efficacy in processing high-dimensional data, such as feature vectors generated from text, Support Vector Machines (SVMs) are commonly used in text classification applications. The initial step in text classification is to convert raw text input into a numerical format that SVMs can operate with. This is typically accomplished using approaches such as bag-of-words or TF-IDF (Term Frequency-Inverse Document Frequency). These approaches convert each text document into a vector, with each dimension representing a distinct word in the entire dataset. Each dimension's value shows the relevance of that word in the specific document.

This optimization problem is about finding the parameters (weights and bias) of the SVM classifier that minimize the specified expression. The expression consists of two components: a term that measures the "hinge loss" or misclassification error and a regularization term that penalizes large values of the weight vector.

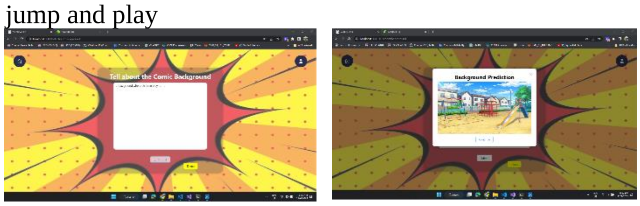
Here's a breakdown of the expression and its components:

1. Hinge Loss Term: The first part of the expression involves computing the sum of the hinge loss for each data point. The hinge loss measures the error incurred by the classifier when it misclassifies a data point. For correctly classified points, the loss is zero, but for misclassified points, the loss increases linearly with the margin of misclassification.
   - For each data point (i):
     - Compute the margin:
       $$1 - y_i * (w^T * x_i - b) \quad (2)$$
     - If the point is misclassified (margin < 1), add the hinge loss to the sum: $max(0, 1 - y_i * (w^T * x_i - b))$
2. Regularization Term: The second part of the expression involves the regularization term, which discourages the weight vector from becoming too large. This helps prevent overfitting.
   - The regularization term is represented by $\lambda * ||w||^2$, where λ is a regularization parameter and ||w||^2 represents the squared Euclidean norm of the weight vector.

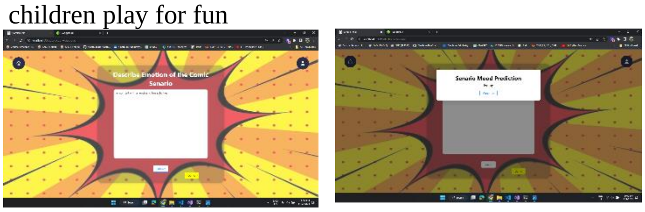
In summary, the equation encapsulates the essence of training a soft-margin SVM classifier. It aims to strike a balance between achieving a wide margin and controlling classification errors while incorporating regularization to prevent overfitting. The below screenshots are from the integrated system.



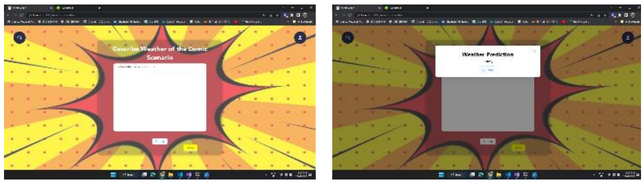
a little girl who wants to jump and play



a playground where children play for fun



a moment with a laugh and fun activities



the beautiful day with the sun rising

Fig. 4. Overall system output

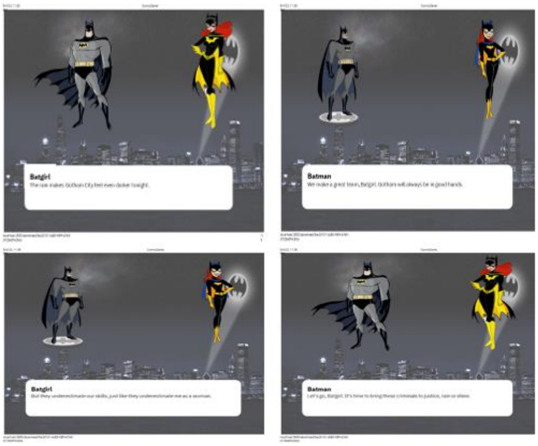


Fig. 5. Generated comic strip

V. CONCLUSION AND FUTURE WORK

In this research endeavour, successfully generated AI-driven comic strips with limited characters, environments, emotions, and dialogues due to resource constraints. The focus on a specific "Batman" narrative allowed us to demonstrate the feasibility of the concept. However, the team acknowledge that this is merely a starting point with the potential for extensive expansion.

As move forward, future works will encompass a broader scope, incorporating a diverse range of characters from various genres, expanded environments, an extensive palette of emotions, and a wider array of dialogues. The constraints encountered in terms of voice cost and environment selection will be mitigated through more comprehensive datasets and improved AI models, facilitating the creation of richer and more dynamic comic narratives.

Additionally, aims to enhance the user experience by integrating a more extensive selection of voices and leveraging advanced voice synthesis technologies. This will lend a unique auditory dimension to the comic strips, further immersing the audience in the storytelling process.

In conclusion, this research has provided a glimpse into the exciting potential of AI-generated comic strips, even within limitations. By successfully demonstrating the creation of comic strips within a focused context, validated the foundational concept. The journey doesn't end here; rather, it marks the inception of a creative evolution. As here embark on future works, the horizon widens, and the possibilities for dynamic, diverse, and engaging AI-generated comic narratives emerge.

REFERENCES

1. Text block segmentation in comic speech bubbles Christophe Rigaud, Nhu-Van Nguyen, JeanChristophe Burie
2. Reddy, A., & Balaji. (2019). Incorporating Visual and Textual Cues in Dialogue Generation: An Application to Comic Strips, 9.
3. Liu, X., Wang, Y., Tang, Z., 2015. A clump splitting based method to localize speech balloons in comics, in: 2015 13th International Conference on Document Analysis and Recognition (ICDAR), IEEE. pp. 901–905.
4. Yamada, M., Budiarto, R., Endo, M., Miyazaki, S., 2004. Comic image decomposition for reading comics on cellular phones. IEICE transactions on information and systems.
5. Understanding Art with AI: Our Research Experience by G Castellano · 2021 — Understanding Art with AI: Our Research Experience. Giovanna Castellano, Gennaro Vessio
6. Wang, Y., Zhou, Y., Liu, D., Tang, Z., 2016. Comic storyboard extraction via edge segment analysis. Multimedia Tools and Applications
7. Tanaka, T., Shoji, K., Toyama, F., Miyamichi, J., 2007. Layout analysis of tree structured scene frames in comic images, in: IJCAI.
8. Su, C.Y., Chang, R.I., Liu, J.C., 2011. Recognizing text elements for svg comic compression and its novel applications, in: 2011 International Conference on Document Analysis and Recognition, IEEE. pp. 1329–1333
9. Ponsard, C., Ramdoyal, R., Dziamski, D., 2012. An ocr-enabled digital comic books viewer, in: International Conference on Computers for Handicapped Persons, Springer. pp. 471–478
10. Burie, J.C., Ogier, J.M., Ho, A.K.N., 2011. Comics page structure analysis based on automatic panel extraction.
11. Wang, Z., Romat, H., Chevalier, F., & Bach, B. (2021). Interactive Data Comics. *ResearchGate*. https://www.researchgate.net/publication/353825597_Interactive_Data_Comics
12. Liang Qiu, Yizhou Zhao, Jinchao Li, Pan Lu, Baolin Peng, Jianfeng Gao, Song-Chun Zhu, 2022 Association for the Advancement of Artificial Intelligence (www.aaai.org)
13. Shibly, K. H., Rahman, S., Dey, S. K., & Shamim, S. H. (2020). Advanced artistic style transfer using deep neural network. In Lecture Notes in Computer Science. https://doi.org/10.1007/978-3-030-52856-0_49
14. Author links open overlay panelChongming Gao a *et al.* (2021) *Advances and challenges in conversational recommender systems: A survey*, *AI Open*. Available at: https://www.sciencedirect.com/science/article/pii/S2666651021000164 (Accessed: 14 August 2023).
15. Ingold, T. (2005). The eye of the storm: visual perception and the weather. *Visual Studies*, *20*(2), 97–104. https://doi.org/10.1080/14725860500243953https://mail.google.com/mail/u/3/ - inbox
16. A. Kusnadi and D. Julio, "Security system with 3 dimensional face recognition using PCA method and neural networks algorithm", 2017 4th International Conference on New Media Studies (CONMEDIA), pp. 152-155, November 2017.
17. Ahmad, A., Amin, M.R.: Bengali word embeddings and it's application in solving document classification problem. In: 2016 19th International Conference on Computer and Information Technology (ICCIT), pp. 425–430. IEEE (2016)
18. L. A. Gatys, A. S. Ecker and M. Bethge, A neural algorithm of artistic style, 2015.
19. D. Rahmawati and M. L. Khodra, "Word2vec semantic representation in multilabel classification for Indonesian news article", 2016 International Conference On Advanced Informatics: Concepts Theory And Application (ICAICTA), 2016.
20. Chiu, J.P., Nichols, E.: Named entity recognition with bidirectional LSTM-CNNs. Trans. Assoc. Comput. Linguist. 4, 357–370 (2016)
21. Figueiredo, F., Rocha, L., Couto, T., Salles, T., Gonçalves, M.A., Meira Jr., W.: Word co-occurrence features for text classification. Inf. Syst. 36(5), 843–858 (2011)
22. Zhang, D., Xu, H., Su, Z., Xu, Y.: Chinese comments sentiment classification based on Word2vec and SVMperf. Expert Syst. Appl. 42(4), 1857–1863 (2015)
23. Goldberg, Y.: Neural network methods for natural language processing. Synth. Lect. Hum. Lang. Technol. 10(1), 117–118 (2017)
24. Sumit, S.H., Hossan, M.Z., Al Muntasir, T., Sourov, T.: Exploring word embedding for Bangla sentiment analysis. In: 2018 International Conference on Bangla Speech and Language Processing (ICBSLP), pp. 1–5. IEEE (2018)
25. Řehůřek, R., Sojka, P.: Software framework for topic modelling with large corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, pp. 45–50. ELRA, Valletta, May 2010. http://is.muni.cz/publication/884893/en