Detection and Classification of Speech Bubbles in Comics Using Convolutional Neural Networks

Alesandro Žužić* ChatGPT†

14/19/2023

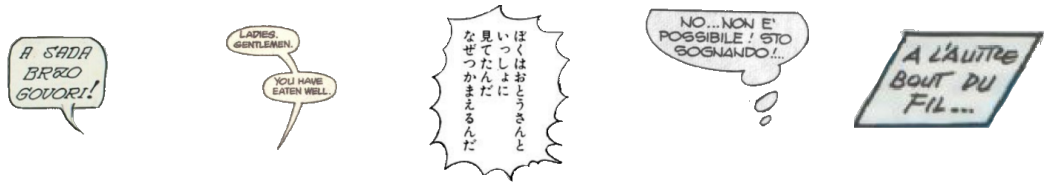
Abstract

This study addresses the classification of speech bubbles in comic book images, a crucial step in automating the analysis of visual narratives. The research employs a custom process for detection and extraction of speech bubbles. This is involved a multi-step process, including image preprocessing, OCR text recognition, grouping of text boxes, binary thresholding, and visualization. Convolutional Neural Network (CNN) model is trained from scratch on a dataset comprising 1000 comic book images, encompassing five distinct speech bubble categories. The model's demonstrates significant test accuracy of approximately 95% after iterative fine-tuning. The study also navigates through challenges posed by unconventional speech bubble styles and limited dataset size. The findings present a robust foundation for advancing the automation of comic book analysis and hold promise for broader applications in visual narrative understanding.

1 Introduction

The precise and effective detection and classification of speech bubbles in comics represent a crucial undertaking in the domain of visual recognition and natural language processing. Speech bubbles serve as a foundational element in the narrative structure of comics, facilitating the communication of dialogue and inner thoughts of characters. This interplay between visual and textual components not only guides the reader's comprehension of the narrative but also imparts a unique aesthetic and storytelling dimension to the medium.

The focus of this research centers on the Detection and Classification of Speech Bubbles in Comics through the application of Convolutional Neural Networks (CNNs). The primary objective is to deploy new detection methodologies, followed by the application of a trained CNN model capable of classifying images into distinct categories, encompassing basic, double, loud, thinking, and narration speech types, as seen in Table 1. The entirety of the dataset has been curated through manual collection, ensuring precision and reliability in the training process.



(a) Basic (b) Double (c) Loud (d) Thinking (e) Narration

Table 1: Speech bubbles categories

The motivation behind this paper stems from the ambition to create new detection model. This involves the application of innovative methodologies, culminating in the development of a completely original CNN model. The aim is to meticulously fine-tune this model to attain an exceptional accuracy threshold surpassing 90%.

*Fakultet informatike, Sveučilište Jurja Dobrile u Puli
†Powered by OpenAI's ChatGPT model

The detection and classification of speech bubbles may seem straightforward at first glance, but it is, in fact, a complex task due to the extensive historical evolution of comics as a medium. With a rich legacy, comics exhibit a wide array of speech bubble styles, presenting a formidable challenge. The dynamic evolution of speech bubbles in both form and function over time adds another layer of intricacy to this task. Furthermore, the varying cultural nuances in the use of speech bubbles across different countries introduce an additional level of complexity to the endeavor.

The structure of this paper is as follows: In the second chapter, an extensive survey of current methodologies within the realm of object recognition and classification employing neural networks is presented. Moving forward, the third chapter offers a comprehensive exposition of the research methodology encompassing data acquisition and processing procedures, the architectural intricacies of the CNN, and the employed dataset augmentation techniques. Following this, the fourth chapter delves into the limitations of the study. Ultimately, the concluding chapter encapsulates a thorough and comprehensive summary of the entire research endeavor, highlighting the achievements, insights gained, and potential avenues for future work.

2 Existing models

The Detection and Classification of Speech Bubbles in Comics using Convolutional Neural Networks (CNNs) constitutes a significant area of interest within the fields of computer vision, natural language processing, and graphic analysis. This section provides an overview of prior research efforts in this domain, highlighting key methodologies and contributions.

Rigaud et al. [1] introduced a pioneering approach for classifying speech balloons in scanned comic books. Their method focused on analyzing contour variations to classify speech balloons into "smooth" (normal speech), "wavy" (thought), or "zigzag" (exclamation). The experiments demonstrated a commendable global accuracy classification of 85.2% on a diverse set of balloons from the eBDtheque dataset.

The eBDtheque dataset [2] is a valuable resource in the field of comics analysis. It comprises a diverse collection of scanned comic book pages, making it suitable for a wide range of experiments and evaluations. This dataset includes various types of speech balloons, each with unique characteristics, which enables researchers to test the robustness of their algorithms across different speech bubble styles.

In a subsequent work, Rigaud and Nguyen [3] addressed text block segmentation within comic speech bubbles. They proposed a domain-specific method capable of detecting single and multiple text block regions inside speech bubbles, which proved beneficial for enhancing OCR transcription and post-processing. This approach yielded highly satisfactory results across a range of bubble styles, encompassing both Latin and non-Latin scripts.

Another significant contribution by Rigaud, Burie, and Ogier [4] centered on text-independent speech balloon segmentation. Recognizing the pivotal role of speech balloons in text/graphic association, they proposed a versatile segmentation method based on color, shape, and topological characteristics of connected components. Evaluation on the eBDtheque and Manga109 datasets yielded F-measure results of 78.24% and 80.04%, respectively.

The Manga109 dataset [5] is another significant resource used in comics-related research. It primarily focuses on manga, which is a distinctive form of comics originating from Japan. This dataset includes a variety of manga pages with speech bubbles, providing researchers with a different cultural perspective. Evaluating algorithms on both the eBDtheque and Manga109 datasets allows for a more comprehensive assessment of their performance across diverse comic book styles.

Dubray and Laubrock [6] presented a method based on deep convolutional neural networks for the automated detection and segmentation of speech balloons in comic books. Their approach, inspired by the U-Net architecture combined with a VGG-16 based encoder, achieved state-of-the-art performance with an F1-score surpassing 0.94. Notably, the model exhibited proficiency in distinguishing speech balloons from captions, even in cases involving wiggly tails, curved corners, and illusory contours.

These studies collectively showcase the evolving landscape of speech bubble detection and classification, highlighting the impact of advanced techniques, including CNNs, on enhancing the understanding and accessibility of comic book content.

3 Methodology

The methodology section encompasses a meticulous approach comprising data collection, speech bubble detection and extraction, and CNN model architecture.

3.1 Data

In this phase, a diverse dataset of comic book pages was meticulously collected, representing various cultural and linguistic backgrounds. This foundational dataset is central to the research.

Notable contributions encompass Japanese classics like "Astro Boy", "Nana", and "Berserk", as well as renowned English-language works including "Batman: Year One", "Garfield", and "Maus". The dataset was further enriched with comics from Croatian, Italian, and French origins, providing distinct cultural and linguistic perspectives. These additions comprised works like "Ahuramazda na Nilu" and "Seoba Hrvata" from Croatian literature, "Dylan Dog" from Italian comics, and iconic titles like "Lucky Luke" and "The Adventures of Tintin" from French comic artistry. For a detailed list, refer to Table 2.

Comic Title	**Author**	**Year**	**Language**
Astro Boy	Osamu Tezuka	1951	Japanese
Slam Dunk	Takehiko Inoue	1990	Japanese
Nana	Ai Yazawa	2000	Japanese
Yotsuba to!	Kiyohiko Azuma	2003	Japanese
Berserk	Kentaro Miura	1989	Japanese
Garfield	Jim Davis	1978	English
Maus	Art Spiegelman	1980	English
Batman: Year One	Frank Miller	1988	English
The Amazing Spider-Man	Ralph Macchio	1999	English
Made in Abyss	Akihito Tsukushi	2012	English
Kaguya-sama: Love Is War	Aka Akasaka	2015	English
Ahuramazda na Nilu	Andrija Maurović	1944	Croatian
Seoba Hrvata	S. R. Žrnovački	1990	Croatian
Lanciostory	Various	1975	Italian
Dylan Dog	Tiziano Sclavi	1986	Italian
Lucky Luke	Maurice De Bevere	1946	French
The Adventures of Tintin	Hergé	1949	French

Table 2: List of Comic Books

The meticulous process of manual extraction involved a systematic approach, entailing the acquisition of copies of each comic in their mother language unless specified otherwise, followed by a page-by-page examination to identify diverse speech bubble instances. Subsequently, each identified bubble was carefully isolated through a rigorous cropping process, wherein precision and attention to detail were paramount. The resulting speech bubble images were further standardized to a uniform size of 256x256 pixels. This painstaking process was iteratively applied, yielding a substantial dataset with 1000 samples (200 samples per category). The dataset encompasses five distinct categories of speech bubbles, each shown in Table 1:

1. Basic: Oval bubble with or without a tail, or square bubble with a tail, Table 3.
2. Double: Two or more connected bubbles, resembling a Venn diagram or connected via tails, Table 4.
3. Loud: Bubble with pointed border featuring oval recesses or pointed borders with angular recesses, Table 5.
4. Thinking: Bubble with multiple circles as a tail, with the shape of the bubble not being a defining factor, Table 6.
5. Narration: Square or rectangular bubble with straight lines and no tail, Table 7.

Table 3: Basic Speech bubbles

Table 4: Double Speech bubbles

Table 5: Loud Speech bubbles

Table 6: Thinking Speech bubbles

Table 7: Narration Speech bubbles

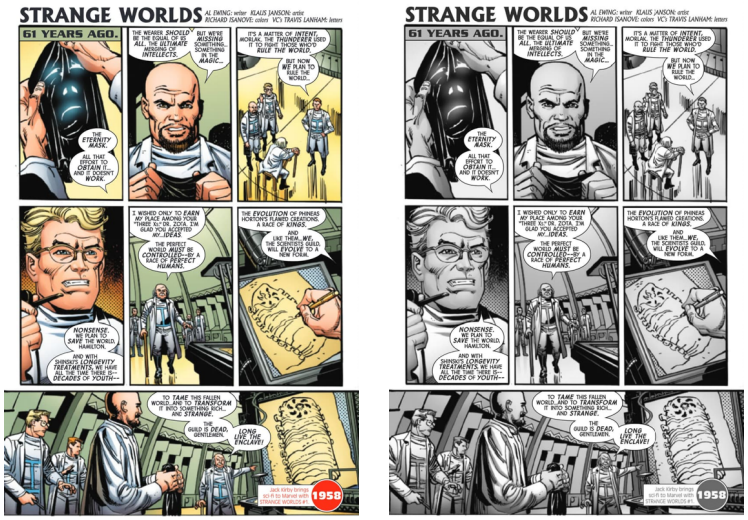
While established datasets such as eBDtheque and Manga109 were available, a deliberate decision was made to construct a bespoke dataset. This choice was motivated by the desire to ensure the dataset's alignment with the specific requirements and nuances of the project, thereby enhancing its suitability for the intended research objectives.

3.2 Detection and Extraction

The next critical phase involves the detection and extraction of speech bubbles from the acquired comic book pages.

3.2.1 Step 1: Image Preprocessing

The process initiates with the loading of the comic book image using a custom-built GUI Python program. The loaded image is preserved in its original form, and a duplicate is created for further processing. The duplicate image is resized to fit within an 800x800 pixel container while maintaining the original aspect ratio. Subsequently, the image is transformed into a binary format using the OpenCV (cv2) library, effectively converting it into a black and white representation, see Table 8.

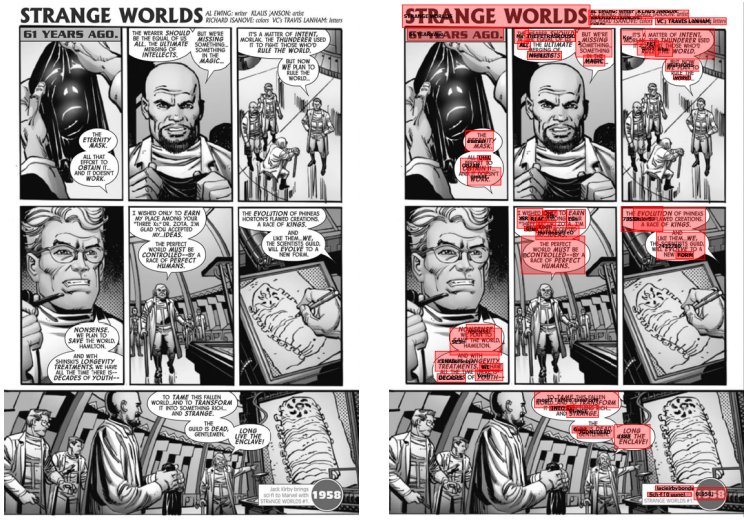


(a) Original (b) Black & White

Table 8: Original and processed image

3.2.2 Step 2: OCR Text Recognition

In this step, Optical Character Recognition (OCR) is employed using the EasyOCR Python library. This approach stands distinct from conventional methods, which often rely on edge detection algorithms. The rationale behind this choice lies in the consistent presence of text within speech bubbles targeted for categorization. Furthermore, OCR technology has demonstrated remarkable proficiency in recognizing text irrespective of font size, style, or color variations. The program provides an interactive dropdown menu within the GUI, enabling users to select the desired language for enhanced text recognition. Upon completion, the OCR process yields a list of identified text segments along with their corresponding bounding box coordinates, see Table 9.

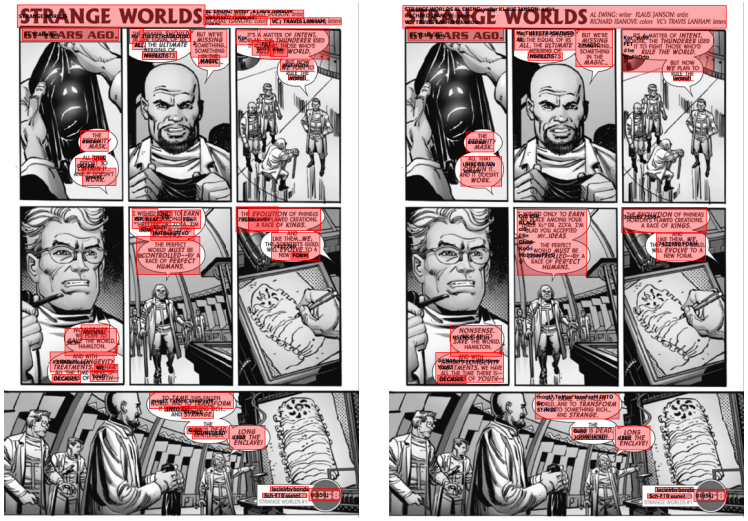


(a) Black & White (b) OCR boxes

Table 9: OCR Text Recognition

3.2.3 Step 3: Grouping of Text Boxes

Since OCR may generate multiple text boxes for a single sentence, it becomes necessary to refine the results. To achieve this, a grouping function is applied. This function iterates through all identified text boxes, consolidating those in close proximity or exhibiting overlap. The outcome is a streamlined collection of combined boxes, ideally corresponding to the number of speech bubbles present, see Table 10.



(a) OCR boxes (b) Grouped boxes

Table 10: Grouping of Text Boxes

3.2.4 Step 4: Binary Thresholding

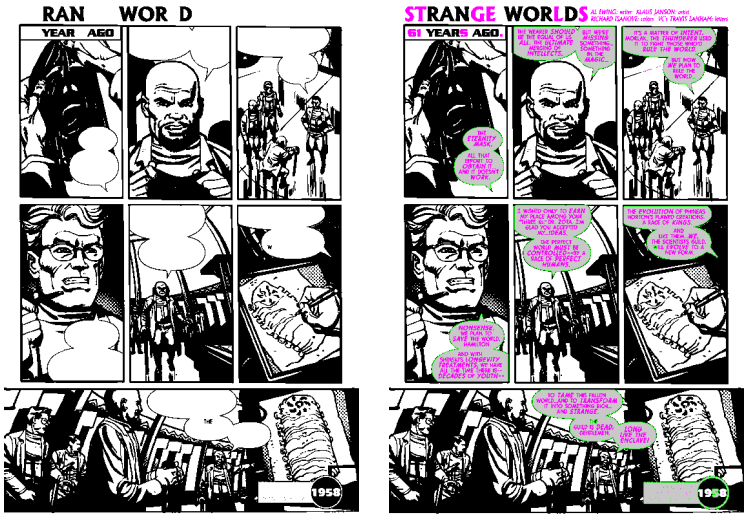
To further enhance the image for subsequent extraction, a binary thresholding operation is employed. This process facilitates the conversion of the image into a high-contrast, purely black and white representation, thereby simplifying the extraction of speech bubbles, see Table 11.

(a) Black & White (b) Binary Thresholding

Table 11: Binary Thresholding

3.2.5 Step 5: Visualization and Speech Bubble Preparation

To visualize the regions identified by OCR, a graphical overlay is applied. Red rectangles are drawn around each detected text segment. Additionally, within each bounding box, the individual characters are obscured, rendering them magenta. Simultaneously, the interior of the speech bubble is filled with a gray hue, while the outline is outlined in green. However, it's important to note that speech bubbles deemed excessively large or small are excluded from this process to mitigate the potential inclusion of extraneous text, see Table 12.



(a) Clean speech bubbles (b) Colored Speech Bubbles

Table 12: Visualization and Speech Bubble Preparation

3.2.6 Step 6: Speech Bubble Extraction

The prepared speech bubbles are extracted from the modified image. Leveraging the known coordinates of each bounding box and the distinctive gray and magenta color scheme, these colors serve as

masks to isolate the speech bubbles from the original image. The extracted speech bubbles are subsequently centered within a 256x256 pixel canvas. If required, resizing is performed to accommodate the predetermined dimensions, as seen in Table 13. The processed speech bubbles are then stored in a designated folder for subsequent classification.

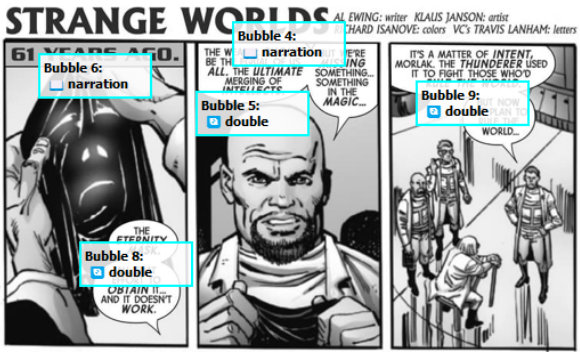
Table 13: Extracted speech bubbles

3.2.7 Step 7: Classification

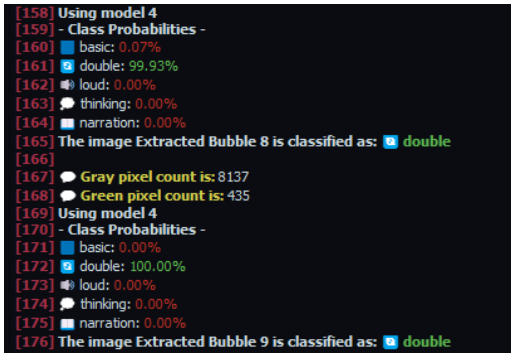
Following the completion of the preceding steps, the extracted speech bubbles are ready for classification. This process involves using a custom-trained Convolutional Neural Network (CNN) model.

The custom-built CNN, which was trained from scratch, employs an output layer with as many units as there are distinct speech bubble categories. This output layer employs a softmax activation function, a widely used technique for multi-class classification tasks. This function assigns probabilities to each category, ensuring that the sum of probabilities across all categories equals 1.

In the graphical user interface (GUI), the categorization results for each bubble are visually displayed. Moreover, in the console, users can access detailed information regarding the percentage likelihood of each bubble belonging to each category, providing a comprehensive assessment of the classification process. This step culminates in the assignment of each speech bubble to its respective category, see Table 14



(a) Categorized speech bubbles



(b) Console category probabilities

Table 14: Categorization result

This methodological approach leverages a combination of image preprocessing, OCR technology, and precise extraction techniques to effectively isolate speech bubbles from comic book pages, thereby establishing a robust foundation for subsequent classification efforts.

3.3 Model Architecture

In this subsection, the development of the Convolutional Neural Network (CNN) model is outlined. To address the task of categorizing speech bubbles from extracted images, a custom convolutional neural network (CNN) was developed using the TensorFlow framework. TensorFlow provides a robust platform for constructing, training, and analyzing neural network models.

The model architecture was programmed in a Jupyter notebook environment, leveraging local computational resources, including a personal NVIDIA GeForce GTX 1080 graphics card with 16 GB of RAM. Jupyter notebooks are interactive computing environments that allow for seamless integration of code, visualizations, and explanatory text, facilitating the development and experimentation process.

This CNN was built from the ground up, without reliance on pre-existing models or extensive adaptation. Each layer was assembled in a way to achieve satisfactory results tailored to the specific requirements of this study.

Convolutional neural networks specialize in the analysis of visual data, particularly images. They excel in tasks such as image classification, object detection, and image segmentation, owing to their capacity to learn intricate features from visual input.

In the context of this problem, the diverse dimensions and shapes of speech bubbles necessitate the recognition of varying spatial patterns and structures. Convolutional Neural Networks (CNNs) exhibit a unique proficiency in learning and discerning these patterns at different scales. Furthermore, CNNs possess several advantageous traits that make them a suitable choice for this task:

1. Multi-Scale Pattern Recognition: CNNs excel at recognizing patterns at different scales within an image. This capability is crucial when dealing with speech bubbles, which can vary significantly in size and shape.

2. Position Invariance: CNNs are capable of identifying patterns irrespective of their position within the image. This characteristic proves invaluable when handling speech bubbles that may vary in position but remain consistent in size.

3. Hierarchical Feature Learning: CNNs automatically learn hierarchical features from the data, starting from basic edges and textures to more complex shapes and structures. This feature learning ability enables them to adapt to the diverse visual characteristics of speech bubbles.

4. Robust to Local Variations: CNNs are robust to local variations and can handle distortions, rotations, and variations in lighting, making them suitable for noisy data like comic book pages, especially after data augmentation is applied.

These inherent properties of CNNs make them a compelling choice for the task of speech bubble classification, where the visual variability and complexity of the speech bubbles demand a model capable of handling diverse patterns and spatial arrangements.

4 Training Model

In this section, the process of training the custom-built Convolutional Neural Network (CNN) for the task of speech bubble classification will be discussed. This critical phase involves fine-tuning the network's parameters to learn the intricate patterns and features that distinguish different speech bubble categories. By the end of this section, the model will be equipped with the ability to accurately categorize speech bubbles based on their visual attributes.

4.1 Step 1 - Checking initial loss and overfiting for small sample

For the initial phase of the training process, a dataset comprising 1000 images categorized into five distinct classes: Basic, Double, Loud, Thinking, and Narration was utilized. In order to assess the performance of the model, a validation split of 0.1 was implemented, ensuring a reliable evaluation procedure. Image size is 256x256 and normalized using Min-Max Scaling (0-1).

The foundational architecture of the initial model consists of seven layers, meticulously designed to capture essential features from the input images. The network commences with two convolutional layers: the first employs 32 filters of size 3x3, activated by Rectified Linear Units (ReLU); the second

employs 64 filters, also with a size of 3x3, and utilizes ReLU activation. These convolutional layers are followed by two max-pooling layers, each with a pool size of 2x2, effectively reducing spatial dimensions.

Subsequently, a crucial step involves flattening the multi-dimensional output, a prerequisite before passing the data to a densely connected layer. Two dense layers follow: the first, featuring 32 units, is activated by ReLU, and the second serves as the output layer. This output layer boasts as many units as there are distinct categories, and employs a softmax activation function for multi-class classification. This function assigns probabilities to each category, with the sum of probabilities equating to 1.

The model is compiled employing the Adam optimizer, instrumental in iteratively adjusting the weights to minimize the loss function. The Sparse Categorical Crossentropy loss function is adopted for training, effectively quantifying the error between predicted and actual labels. Furthermore, during training, the accuracy metric is monitored, offering a comprehensive insight into the model's performance.

Upon the inception of the initial model, an essential step involves the evaluation of its performance prior to any training iterations. The initial loss, as determined through assessment, stands at approximately 160%. This metric serves as an initial benchmark, reflecting the magnitude of error between predicted and actual labels.

In tandem with this, the initial accuracy of the model registers at around 23%. This figure aligns with expectations, given that the model has yet to undergo any training.

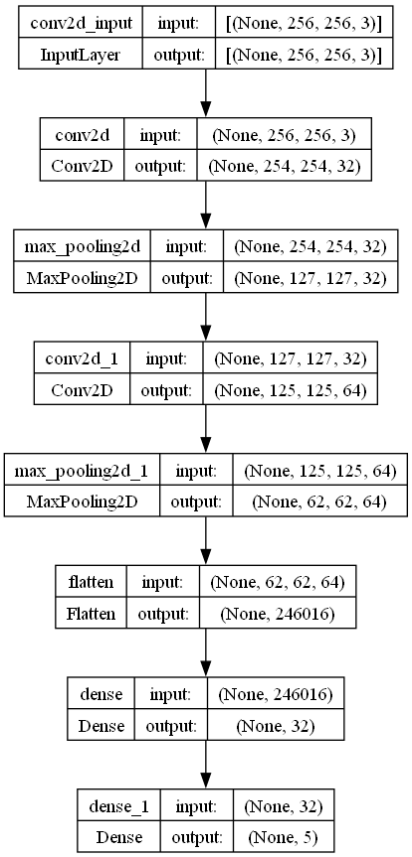


Figure 1: Model 1 - summary

After obtaining the initial loss, the model was subsequently trained.

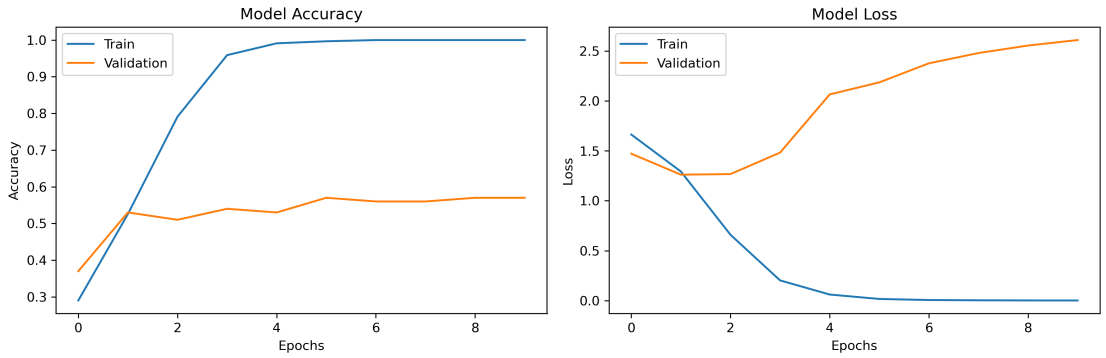


Figure 2: Model 1 - plot

The outcomes are as follows:

- Train Accuracy ascended to 100% and consistently maintained this performance throughout the training process, which is expected behaviour.
- Test Accuracy exhibited an initial value of approximately 37%, experienced a subsequent rise to around 57%, and then stabilized.
- Train Loss decreased to nearly 0%, which is expected behaviour.
- Test Loss commenced at approximately 160%, reached a plateau, and subsequently increased to around 260%. This trend suggests a potential issue of overfitting, particularly noticeable in the significant gaps between the lines in both graphs.

4.2 Step 2 - Using augmented data and reducing overfitting

In this step, the aim is to address overfitting by incorporating augmented data into the training process. Furthermore, supplementary layers will be introduced to enhance model stability and counteract overfitting tendencies.

4.2.1 Data Augmentation

The augmentation of data involves the deliberate introduction of diverse transformations to the existing dataset, amplifying its variability and robustness. This process encompasses a range of techniques, including controlled rotation of images within a 15-degree scope, deliberate horizontal and vertical shifts, application of shear transformations, controlled zooming, and both horizontal and vertical flipping, see Figure 15.

Augmentation techniques involving adjustments in brightness, black box blocking, and cropping were deliberately omitted from the process. This cautious approach was adopted due to the potential risk of compromising critical image features, particularly in cases of thinking speech bubble circles with tails. There was also a concern that such techniques could inadvertently transform one speech bubble into another, thereby undermining the integrity of the dataset. This decision was made to preserve the authenticity and accuracy of the dataset for training purposes.

Table 15: Augmented Data

4.2.2 2. Model

In the second model iteration, measures were taken to mitigate overfitting. These included the implementation of L2 regularization, which contributes to a more balanced distribution of the model's weights, thereby reducing its susceptibility to noise in the data. Additionally, a dropout layer was introduced to curtail overfitting by minimizing reliance on specific neurons in the network.

- Three batch normalization layers were incorporated to stabilize and enhance the training process by normalizing inputs for each mini-batch.
- The model's dense layers consist of a unit-connected layer with 32 units, employing ReLU activation and featuring L2 regularization with a regularization strength of 0.001.
- A dropout layer was introduced with a dropout rate of 0.05 to further counter overfitting.
- The Adam optimizer was employed, utilizing a learning rate of 0.005 to adjust the weights in the network for minimizing the loss function.

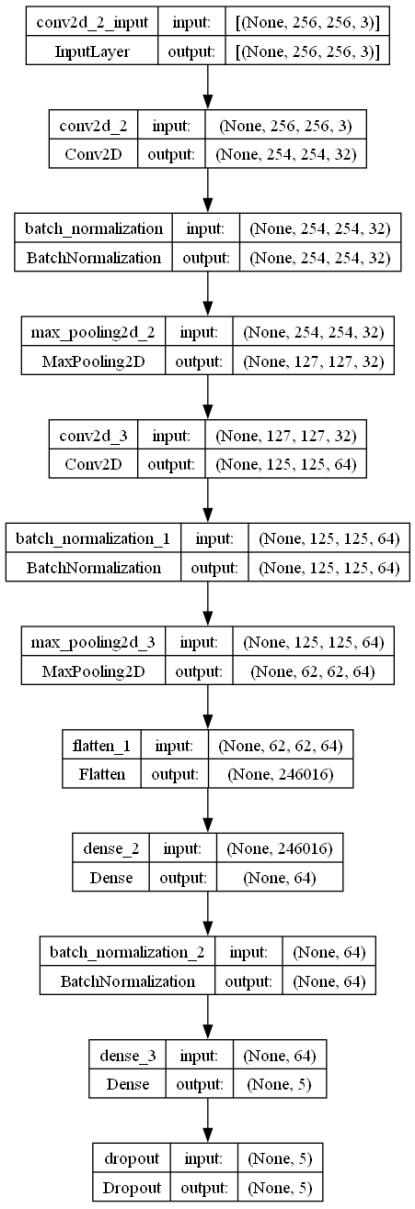


Figure 3: Model 2 - summary

Following the augmentation of the dataset, which expanded it to 3000 images with a 0.1 validation split, the second model underwent training.

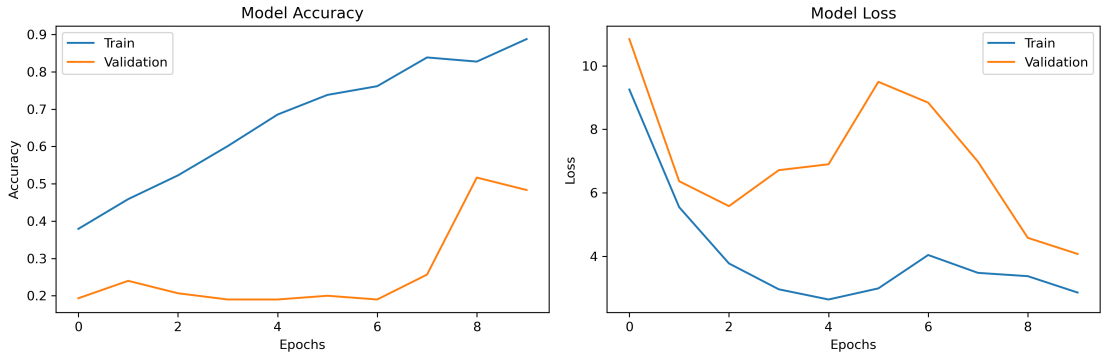


Figure 4: Model 2 - plot

The results indicate a noticeable improvement in training accuracy, which gradually ascended to approximately 88%. This suggests a reduction in overfitting compared to the initial model. Conversely, the test accuracy exhibited a more nuanced pattern. Initially starting at around 19%, it stabilized before experiencing a subsequent increase, ultimately reaching approximately 51%.

Regarding the loss metrics, the training loss diminished considerably, reaching approximately 285%. In contrast, the test loss exhibited a fluctuating trajectory. Commencing at approximately 1084%, it underwent a period of reduction followed by an increase to around 949%. However, in the latter stages of training, the test loss experienced a rapid decrease, progressively closing the gap between training and testing losses. This culminated in a test loss of approximately 407%.

Despite the presence of substantial gaps between the lines in both graphs, it is evident that toward the conclusion of the training process, these gaps began to diminish.

4.3 Step 3 - Training longer and tweaking

In the third model iteration, several adjustments were made to further refine the training process. Specifically, the dropout rate was increased to 0.1 to enhance regularization. Additionally, during compilation, the learning rate was reduced to 0.001 to promote a more controlled adjustment of weights in the network. Moreover, the number of epochs was extended to 20 to allow for more extensive training.

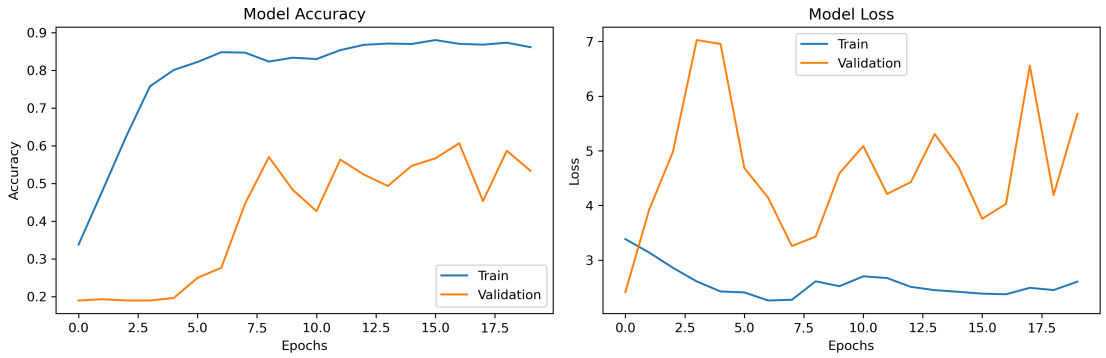


Figure 5: Model 3 - plot

Upon training, the results exhibited discernible changes. The training accuracy saw a gradual increase, reaching a peak of approximately 86% after a steady climb from around 80%. Conversely, the test accuracy demonstrated an improvement. Initially starting at approximately 22%, it experienced a notable enhancement, ultimately reaching around 60%. However, it's worth noting that the test accuracy line displayed some fluctuations, indicating that the learning rate in later epochs might be too high.

Concerning the loss metrics, the training loss diminished significantly, achieving a value of approximately 26%. In contrast, the test loss exhibited a different pattern. While it reached a peak of 567%, the line displayed some oscillation, suggesting that the learning rate might be too large.

Despite these adjustments, notable gaps between the lines in both accuracy and loss graphs still persist. This indicates the presence of substantial overfitting. Additionally, the behavior of the test loss line suggests that the learning rate may be too high, especially in later epochs.

Given the current observations, it appears imperative to augment the model's complexity by incorporating additional layers.

4.4 Step 4 - Adding more layers

The fourth iteration encompassed the integration of 23 layers, designed to enhance the model's capacity for discerning intricate features. This augmentation involved the inclusion of six convolutional layers, each equipped with a diverse set of filters and activation functions tailored to the unique characteristics of the data.

- convolutional layer with 32 filters of size 3x3, using ReLU activation function
- convolutional layer with 32 filters of size 5x5, using ReLU activation function
- convolutional layer with 32 filters of size 7x7, using ReLU activation function
- convolutional layer with 64 filters of size 3x3, using ReLU activation function
- convolutional layer with 64 filters of size 5x5, using ReLU activation function
- convolutional layer with 128 filters of size 3x3, using ReLU activation function

Moreover, seven batch normalization layers were introduced to stabilize and optimize the training process by standardizing inputs within each mini-batch. To further reduce overfitting, six max pooling layers were positioned after batch normalization layers to diminish spatial dimensions. Dropout layer rate increased to 0.2, acting as an additional safeguard against overfitting, effectively enhancing the model's ability to generalize from the training data to unseen examples.

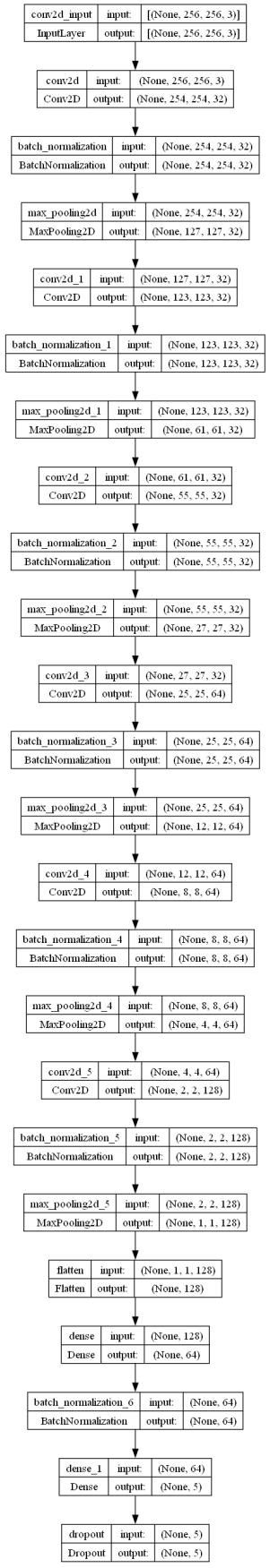


Figure 6: Model 4 - summary

The model's training regimen was complemented with adaptive measures. A learning rate scheduler was integrated, facilitating a gradual reduction of the learning rate by a factor of 0.9 following the completion of each epoch. This adjustment addressed a prior observation indicating that the learning rate might have been too high in later stages of training, underscoring the importance of fine-tuning this parameter for optimal performance. Moreover, an early stopping mechanism was instituted, designed to halt the training process should the validation loss exhibit no improvement for ten consecutive epochs. This strategic addition was conceived to optimize training efficiency and performance.

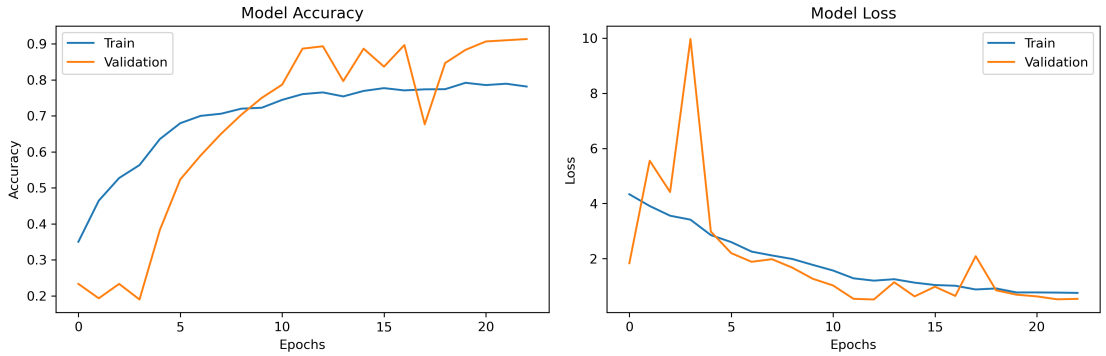


Figure 7: Model 4 - plot

The outcomes for Model 4 unveil noteworthy progress in several key aspects. The training accuracy demonstrates a steady ascent, ultimately stabilizing at approximately 78%. In contrast, the test accuracy embarks on a more remarkable trajectory, commencing at around 18% and ascending to an impressive 91% - a substantial enhancement.

On the loss front, both training and test losses exhibit positive trends. The training loss diminishes to approximately 75%, while the test loss registers a reduction to around 53%. This convergence of loss lines without any discernible gap is an encouraging sign, indicating a closer alignment between predicted and actual values.

However, it is worth noting that the accuracy metrics hint at potential underfitting. In light of this observation, a prudent next step would be to reevaluate the regularization strategies employed, with an eye toward potential adjustments to strike a more optimal balance.

4.5 Step 5 - Final improvements and fine-tuning

In this last step, several refinements were introduced to bolster the model's performance. Notable adjustments included the transition from ReLU to LeakyReLU activation functions, a proactive measure to mitigate the "dying ReLU" issue. In the context of speech bubble classification, this can be particularly problematic. Since speech bubbles exhibit a wide variety of shapes, sizes, and textures, the neurons responsible for recognizing specific features may encounter challenges. If a ReLU neuron becomes "dead," it may fail to recognize critical patterns in the speech bubbles, potentially leading to misclassifications.

Kernel initializers were introduced to the convolutional layers, a crucial factor in setting the initial weights of each layer. Specifically, the LeakyReLU activation function now employs the Lecun Normal kernel initializer, while the ReLU activation function benefits from the He Normal kernel initializer.

The dropout layer rate was reduced to 0.075. This adjustment was prompted by the observation of signs indicating potential underfitting in previous results.

23 layers:

- 6 convolutional layers:
  - convolutional layer with 32 filters of size 3x3, using ReLU activation function and kernel initializer He Normal
  - convolutional layer with 32 filters of size 5x5, using ReLU activation function and kernel initializer He Normal

  - convolutional layer with 32 filters of size 7x7, using LeakyReLU activation function and kernel initializer Lecun Normal
  - convolutional layer with 64 filters of size 3x3, using ReLU activation function
  - convolutional layer with 64 filters of size 5x5, using ReLU activation function
  - convolutional layer with 128 filters of size 3x3, using LeakyReLU activation function and kernel initializer Lecun Normal
- 7 batch normalization to stabilize and improve the training process by normalizing the inputs for each mini-batch
- 6 max pooling layers with a pool size of 2x2, reducing the spatial dimensions
- 1 flatten layer
- 2 dense layers:
  - densely connected layer with 32 units, ReLU activation and L2 regularization with a regularization strength of 0.001
  - output layer, has as many units as there are categories with a softmax activation function
- 1 dropout layer with dropout rate of 0.075

The learning rate was augmented to 0.0025. This alteration was implemented with the expectation that the learning rate would decrease naturally over time, enhancing the model's capacity for intricate learning and adaptation.

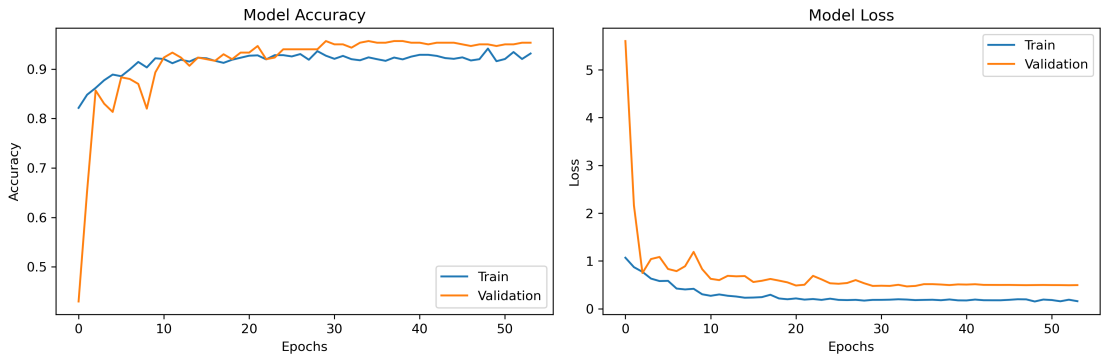


Figure 8: Model 5 - plot

The refined model, denoted as Model 5, exhibited notable advancements in performance:

- Train Accuracy: Elevated to approximately 93
- Test Accuracy: Demonstrated a substantial improvement, reaching approximately 95
- Train Loss: Reduced significantly to around 15
- Test Loss: Decreased notably to approximately 49

These results signify a high degree of alignment between the training and testing outcomes, indicating that the model has achieved a commendable level of proficiency in its classification task.

Further refinement and fine-tuning yielded diminishing returns, characterized by a decrease in accuracy and an increase in loss. This phenomenon suggests a potential bottlenecking effect arising from the limitations within the training data. It is conceivable that further enhancements in model performance may necessitate an augmentation of the dataset or exploration of more advanced training techniques.

5 Limitations

This study, while providing valuable insights into speech bubble classification, is not without its limitations:

1. Limited Dataset: Although the dataset utilized in this study comprised 1000 images, it may be considered relatively small when compared to the extensive range of speech bubbles present in comics worldwide. Moreover, the dataset primarily represents specific years and languages, omitting notable contributions from Korean manhwa.

2. Algorithmic Constraints in Bubble Extraction: While the speech bubble extraction algorithm is effective in most cases, it may encounter difficulties with unconventional speech bubble features. Notably, it may struggle with extracting the circular tail associated with thinking speech bubbles. Additionally, the model may face challenges with uniquely stylized speech bubbles, such as those combining multiple shapes, utilizing unconventional outlines, employing distinct designs for individual characters, employing inverted color schemes, adding symbol or drawings inside speech bubbles, making text larger than bubbles, Venn bubbles, stacking bubbles, or introducing entirely novel bubble designs, see Table 16

Table 16: Unique speech bubbles

3. Schrödinger's Bubble Phenomenon: The study acknowledges the existence of what can be termed "Schrodingers Bubbles". These encompass instances where artists employ speech bubbles in unorthodox or inventive ways, such as amalgamating different bubble shapes. This variability introduces complexity and potential limitations to the extraction process, see Table 17

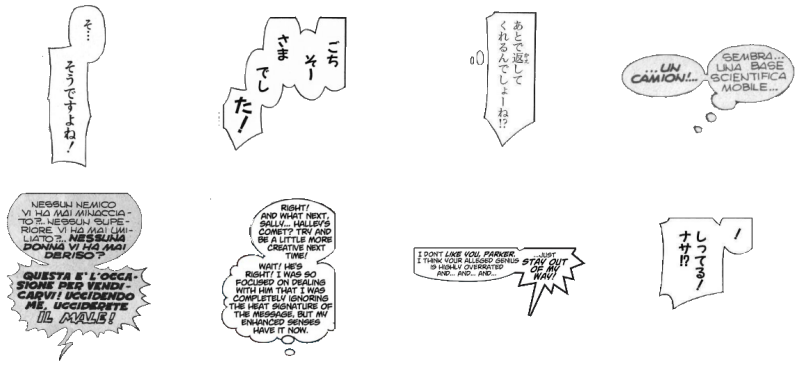


Table 17: Schrödinger's speech bubbles

These limitations provide avenues for future research to refine and expand upon the methodologies employed in this study, ensuring more comprehensive and adaptable approaches to speech bubble classification in comics.

6 Conclusion

This study undertook the intricate task of classifying speech bubbles in comic book images, employing Convolutional Neural Networks (CNNs) as the core methodology. The endeavor encompassed a comprehensive process, spanning from data acquisition and extraction to pre-processing, model development, and extensive fine-tuning.

The initial phase of data extraction and pre-processing played a pivotal role in shaping the dataset. By incorporating comics from Croatian, Italian, and French origins, we enriched the dataset with diverse cultural and linguistic dimensions. The painstaking process of manual extraction involved a systematic approach, acquiring copies of each comic, followed by a page-by-page examination to identify diverse speech bubble instances. Subsequently, each identified bubble was carefully isolated through a rigorous cropping process, wherein precision and attention to detail were paramount. The resulting speech bubble images were further standardized to a uniform size of 256x256 pixels.

The subsequent phase focused on the detection and extraction of speech bubbles. This involved a multi-step process, including image preprocessing, OCR text recognition, grouping of text boxes, binary thresholding, and visualization. The extracted speech bubbles were further processed and prepared for classification.

In the realm of model development, a bespoke CNN architecture was meticulously crafted to cater specifically to the intricacies of speech bubble classification. Numerous iterations and augmentations were undertaken to combat overfitting and enhance the model's capacity to discern the wide array of speech bubble patterns.

The models displayed varying degrees of success, with each iteration demonstrating improvements over its predecessor. Model 5, in particular, exhibited impressive accuracy of 95%, indicating a high degree of proficiency in speech bubble classification.

However, it was observed that further fine-tuning and model enhancements began to yield diminishing returns. This led to the hypothesis that the training data itself may be a limiting factor in achieving even higher accuracy. This warrants consideration for future work, potentially through the expansion of the dataset or exploration of advanced training techniques.

In conclusion, this study not only showcases the potential of Convolutional Neural Networks in comic book analysis but also emphasizes the critical importance of data extraction and pre-processing. The developed models demonstrate commendable proficiency in speech bubble extraction and classification, establishing a robust foundation for future research and applications in this intriguing field. The methodologies and insights gained from this endeavor provide valuable contributions to the intersection of computer vision and comics studies.

References

[1] Christophe Rigaud, Dimosthenis Karatzas, Jean-Christophe Burie, and Jean-Marc Ogier. Adaptive contour classification of comics speech balloons. In Bart Lamiroy and Jean-Marc Ogier, editors, *Graphics Recognition. Current Trends and Challenges*, pages 53–62, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[2] Clément Guérin, Christophe Rigaud, Antoine Mercier, Farid Ammar-Boudjelal, Karell Bertet, Alain Bouju, Jean-Christophe Burie, Georges Louis, Jean-Marc Ogier, and Arnaud Revel. ebdtheque: A representative database of comics. pages 1145–1149, 08 2013.

[3] Christophe Rigaud, Nhu-Van Nguyen, and Jean-Christophe Burie. Text block segmentation in comic speech bubbles. In Alberto Del Bimbo, Rita Cucchiara, Stan Sclaroff, Giovanni Maria Farinella, Tao Mei, Marco Bertini, Hugo Jair Escalante, and Roberto Vezzani, editors, *Pattern Recognition. ICPR International Workshops and Challenges*, pages 250–261, Cham, 2021. Springer International Publishing.

[4] Christophe Rigaud, Jean-Christophe Burie, and Jean-Marc Ogier. Text-independent speech balloon segmentation for comics and manga. In Bart Lamiroy and Rafael Dueire Lins, editors, *Graphic Recognition. Current Trends and Challenges*, pages 133–147, Cham, 2017. Springer International Publishing.

[5] Azuma Fujimoto, Toru Ogawa, Kazuyoshi Yamamoto, Yusuke Matsui, Toshihiko Yamasaki, and Kiyoharu Aizawa. Manga109 dataset and creation of metadata. In *Proceedings of the 1st International Workshop on CoMics ANalysis, Processing and Understanding*, MANPU '16, New York, NY, USA, 2016. Association for Computing Machinery.

[6] David Dubray and Jochen Laubrock. Deep cnn-based speech balloon detection and segmentation for comic books. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1237–1243, 2019.