Solutions for Integrating Charts into Email Communications

Venkata Padma Kumar Vemuri

padma.vemuri@gmail.com
Santa Clara, CA

Abstract:
The integration of visual charts within email communications provides significant advantages in capturing recipient attention, simplifying complex datasets, and enhancing the clarity and memorability of key messages. This research report explores various reliable techniques for incorporating charts into emails, highlighting embedding static images as the most universally compatible method. Utilizing HTML image tags ensures broad compatibility across email platforms, while leveraging online chart generation APIs, such as QuickChart.io and Image-Charts, simplifies the creation and embedding process by managing server-side rendering complexities.
The choice of image format is crucial, with PNG recommended for detailed graphics and clarity preservation, JPEG beneficial for complex, photograph-like charts prioritizing smaller file sizes, and GIF suitable for simple animations or graphics with limited colors. Advanced considerations, including mobile optimization, cross-client compatibility, and accessibility through effective alternative text descriptions, are discussed as best practices to enhance user experience and inclusivity.
Finally, the limitations of embedding interactive charts directly in emails due to email client restrictions are acknowledged, recommending alternative approaches like linking static chart images to interactive online versions. Overall, the report provides comprehensive guidance for effectively employing visual charts in email communications, ensuring messages stand out, engage recipients, and clearly convey critical data and insights.

Keywords: Email Communication, Visual Data Integration, Chart Embedding, Static Images, Online Chart APIs, QuickChart, Image-Charts, Cross-Client Compatibility, Mobile Optimization, Accessibility, Alternative Text

1. Introduction: The Role of Visual Data in Email Communication

This research report explores various solutions for integrating charts into email communications. In the contemporary digital landscape, the ability to effectively capture and retain the attention of email recipients is paramount. One potent method for achieving this is the strategic incorporation of visual data, particularly charts and graphs, directly within the body of email communications. These visual representations serve as powerful tools for simplifying intricate datasets, enabling recipients to grasp key insights and trends with greater speed and clarity. The human brain processes visual information far more efficiently than it does large blocks of text, making charts and graphs an invaluable asset for conveying complex information in an easily digestible format. By breaking up dense textual content, visuals facilitate a more rapid understanding of the underlying data and significantly enhance recipient engagement with the message.

The sheer volume of emails individuals receive daily underscores the necessity of employing methods that allow messages to stand out and quickly convey their intended meaning. Visual elements, such as charts, offer an immediate point of differentiation, drawing the reader's eye and prompting further engagement with the content. This is particularly crucial in professional settings where time is often limited, and recipients need to quickly assess the information presented to them. The integration of charts into emails can transform standard communications into impactful messages, ensuring that key data points and analytical findings are not only seen but also readily understood and remembered.

2. Fundamental Techniques for Sending Emails with Charts

2.1. Embedding Charts as Static Images: A Comprehensive Overview

The most universally compatible and reliable method for displaying charts within the body of an email involves embedding them as static images. This technique leverages the fundamental way email clients handle visual content, primarily through the use of the HTML <img> tag. Email clients are designed to render standard HTML, and the <img> tag is a core element for displaying images, ensuring broad compatibility across various email platforms and devices.

A practical approach to obtaining a chart image for embedding involves utilizing online chart generation services. These platforms, such as QuickChart.io, handle the complexities of rendering charts from provided data or configurations into standard image formats. By providing chart specifications, often in JSON format adhering to the Chart.js syntax, as URL parameters to the service's API, a user can retrieve a URL that points directly to a rendered chart image. This URL can then be seamlessly integrated into the body of an email using the syntax. This method simplifies the process, abstracting away the need for users to manage server-side rendering infrastructure.

Another common workflow involves fetching a chart as an image from a specific platform, such as a data visualization tool or a cloud service like Qlik Cloud. This image can then be inserted into the email body, formatted either as markdown or directly as HTML. In some scenarios, it might be necessary to encode the image data into a base64 string. Base64 encoding allows the image data to be directly embedded within the HTML of the email as a long string of characters, eliminating the need for an external URL reference. While this approach can be convenient for creating self-contained emails, it is important to note that base64 encoding can significantly increase the overall size of the email, potentially impacting its deliverability and loading times for recipients.

The necessity of using static images stems from the inherent limitations of email client security protocols. Most email clients are designed to strip out or block JavaScript code to prevent malicious scripts and ensure a consistent user experience. Since many interactive charts rely on JavaScript for their functionality, embedding them directly into an email body is generally not feasible. Therefore, rendering charts as non-interactive image formats becomes the most reliable option for visual data representation in this context.

An alternative technique for embedding images, including charts, is the use of CID (Content-ID). This method involves attaching the image file to the email and then referencing it within the email body using a unique identifier. The syntax for this typically looks like <img src="cid:unique_identifier">. While this approach can ensure that the image is displayed even if the recipient's email client blocks external images by default, some testing has indicated potential limitations, such as only being able to reliably embed a single image using this method. Therefore, while CID embedding offers good compatibility, its constraints might make it unsuitable for emails containing multiple charts.

2.2. Leveraging Online Chart Generation APIs for Email Integration

A growing trend in the realm of email communication with visual data is the utilization of specialized online services that offer APIs for generating chart images on demand. These services abstract the complexities associated with server-side rendering, providing a streamlined way for users to create and embed charts in emails with minimal coding effort. They manage the underlying infrastructure and software required to generate chart images in various formats, freeing users from the need to set up and maintain their own rendering environments.

The process typically involves constructing an API request that includes the desired chart configuration. This configuration is often provided in JSON format and follows the syntax of popular web-based charting libraries like Chart.js. The chart configuration is then passed as a parameter, often named chart or c, within the URL of the API endpoint provided by the service, such as https://quickchart.io/chart?c={chart_configuration}. Upon receiving this request, the API renders the chart as an image and returns a URL that points to this image. This resulting image URL can then be directly embedded into the body of an email using the standard HTML <img> tag, as in .

The reliance on standard chart configuration formats like Chart.js makes these APIs highly accessible to a wide range of users who are already familiar with web-based charting libraries. Users can leverage their existing knowledge and easily translate charts designed for web applications into a format suitable for embedding in emails. Furthermore, the use of URLs for embedding chart images simplifies the email creation

process. These URLs can be dynamically generated based on data retrieved from various sources or based on user input, opening up possibilities for automating the creation of personalized emails that include up-to-date and relevant charts.

Beyond QuickChart.io, other services like Image-Charts also exist, offering similar functionalities for generating chart images via an API. These services often require the creation of a URL string that includes various parameters defining the chart type, data, labels, and appearance. Once this URL is constructed, it can be used within an <img> tag to embed the generated chart image directly into an email. The presence of multiple competing services in this space provides users with a range of options to consider based on specific features, pricing models, and ease of integration with their existing systems.

Notably, these online chart generation APIs can serve as effective intermediaries for platforms that primarily produce interactive charts, such as Google Charts. Since embedding interactive charts directly in emails is generally not supported, services like QuickChart.io can be used to render static image versions of these charts, providing a practical workaround for users who need to include Google Charts in their email communications.

A crucial technical detail to consider when using these APIs is the importance of properly URL encoding the chart configuration, especially when it is provided in JSON format. URL encoding ensures that any special characters or spaces within the JSON configuration are correctly interpreted by the API service, preventing errors in the chart rendering process and ensuring the reliable generation of the desired chart image. Most programming languages provide built-in functions for URL encoding, simplifying this step for developers.

3. Choosing the Right Image Format for Email Charts: A Comparative Analysis (PNG, JPEG, GIF)

When embedding charts as static images in emails, the choice of image format can significantly impact the visual quality, file size, and overall effectiveness of the communication. The three most common formats used for this purpose are PNG, JPEG, and GIF.

JPEG (or JPG) is a widely supported format known for its ability to achieve significant file size reduction through lossy compression. This makes it particularly suitable for photographs and complex charts with many colors where minimizing file size is a priority, as smaller files lead to faster email loading times However, the lossy compression inherent in JPEG can result in a degradation of image quality, especially noticeable in images with sharp lines, text, or flat areas of color, which are common in charts Additionally, JPEG does not support image transparency.

PNG (Portable Network Graphics) employs lossless compression, ensuring that no image data is lost during the compression process. This makes PNG an excellent choice for detailed graphics, charts containing text or fine lines, and logos, as it preserves image clarity and sharpness PNG also supports transparency, allowing charts to be seamlessly overlaid on email backgrounds However, PNG files generally tend to be larger in size compared to JPEGs, which might be a consideration for emails with multiple images or recipients with slower internet connections

GIF (Graphics Interchange Format) is best suited for simple graphics, animations, and images with a limited color palette of up to56 colors. It supports transparency and can result in smaller file sizes than PNG for images with few colors A key advantage of GIF is its ability to support simple animations, which can sometimes be used to highlight specific data points in a chart. However, the limited color palette of GIF makes it less ideal for complex charts with gradients or a wide range of colors, as these might appear flat or grainy.

The optimal choice of image format depends on the specific characteristics of the chart and the desired outcome. For charts with detailed information, text, and sharp lines, PNG is generally the preferred format due to its lossless compression and support for transparency If the chart is relatively simple with few colors or if animation is desired, GIF can be a good option, offering smaller file sizes JPEG might be considered for very complex charts that resemble photographs where a smaller file size is crucial and some loss of quality in fine details is acceptable

Table: Image Format Comparison for Email Charts

Format	Compression Type	Best Suited For	Transparency Support	Animation Support	File Size	Considerations
PNG	Lossless	Detailed Graphics, Charts with Text, Logos	Yes	No	Larger	Excellent quality, preserves sharpness
JPEG	Lossy	Photographs, Complex Charts (if quality loss is acceptable)	No	No	Smaller	Good for photos, quality can degrade, not ideal for text
GIF	Lossless (for indexed colors)	Simple Graphics, Logos, Animations	Yes	Yes	Smaller (for limited colors)	Limited color palette (256 colors)

4. Advanced Considerations and Best Practices

4.1. Optimizing Charts for Email Display and File Size

To ensure that charts embedded in emails are effective and perform well, several optimization considerations should be taken into account. Chart dimensions are critical, especially given the prevalence of mobile email viewing. Overly wide or complex charts designed for desktop viewing can become illegible when scaled down for smaller mobile screens. A desktop width limit of around 550 pixels is often cited as a guideline. Designing charts with a mobile-first approach, or providing a thumbnail that links to a larger, more detailed version hosted online, can improve the user experience.

Optimizing charts for email display and file size involves several considerations, including the design of visualization snippets, the impact of display size, and the principles of graphical excellence[1][2]. Visualization snippets, as proposed by Oppermann and Munzner, can be compressed into representative previews that maintain high information density while adhering to a pixel budget, which is crucial for email displays where space is limited. Jakobsen and Hornbæk's research highlights the importance of considering display size and scale ratio, suggesting that smaller displays, like those in emails, may not benefit from large-scale visualizations, and that performance can be optimized by adjusting the information space relative to the display size. Stengel et al[3]. emphasize the need for high data density and a low ink-to-data ratio, which can be achieved by avoiding unnecessary elements such as shading and gridlines, thus reducing file size and enhancing clarity[4]. Additionally,Many advocate for decluttering and focusing techniques, which involve removing non-critical elements and highlighting key data points, thereby improving aesthetics and clarity, which are essential for effective communication in emails[3]. Finally, Lu et al. demonstrate the utility of rich visual elements in email visualization systems, which can help users understand complex datasets without overwhelming them, thus balancing detail with simplicity. By integrating these strategies, charts can be optimized for email display, ensuring they are both informative and efficient in terms of file size.[5]

4.2. Ensuring Cross-Client Compatibility

The rendering of HTML emails and embedded images can vary across different email clients, including desktop applications like Outlook and Thunderbird, and webmail services like Gmail. To mitigate potential issues, it is best to adhere to standard HTML practices for embedding images, primarily using the <img> tag with a direct URL or base64 encoded data. Choosing widely supported image formats such as JPEG, PNG, and GIF is also crucial for broad compatibility While embedding images using CID (Content-ID) generally works well across most email clients, there might be limitations, such as the number of images that can be embedded this way. It's also worth noting that while PNG has widespread support, older email clients like early versions of Lotus Notes might not render them correctly Complex or non-standard HTML generated by certain tools might also lead to inconsistent rendering in some email clients, such as Outlook. The most

effective way to ensure cross-client compatibility is to use simple, well-supported HTML structures, choose appropriate image formats, and thoroughly test emails across various popular email clients before sending them to a large audience.

4.3. Accessibility and Alternative Text for Charts

Ensuring that emails with embedded charts are accessible to all recipients, including those with visual impairments, is a critical aspect of inclusive communication. A fundamental best practice for achieving this is to provide descriptive and relevant alternative text (alt text) for each image, including charts, embedded in the email. The alt text serves two primary purposes: it allows screen reader software to convey the content and purpose of the image to visually impaired users, and it provides a fallback description in cases where the image fails to load for any reason. When writing alt text for a chart, it should be concise yet informative, summarizing the key takeaways and trends presented in the visual representation. This ensures that all recipients can understand the essential information being communicated, regardless of their visual abilities.

Accessibility of charts for visually impaired individuals, particularly through alternative text, is a multifaceted issue that has been explored across various studies. During the Covid-19 pandemic, the accessibility of statistical charts for people with low vision was found to be lacking, with issues such as poor text alternatives and insufficient contrast being prevalent in charts from major health organizations and governments[6]. To address these challenges, several approaches have been proposed, including the use of natural language descriptions to enhance the accessibility of visualizations[7]. A four-level model of semantic content has been developed to evaluate the effectiveness of these descriptions, highlighting the importance of tailoring content to reader preferences. Additionally, methods have been suggested to convert graphical information into audio formats, providing vision-impaired users with textual summaries of charts through text-to-speech software. Systems like TAIG and SIGHT have been developed to generate textual summaries of information graphics, allowing users to access the core content and respond to follow-up questions for further detail[8]. Despite these advancements, challenges remain, particularly in decoding visual data from chart images using AI approaches, which require improvements in reasoning and data extraction from raster images. The accessibility of visual content in biomedical journals is also problematic, with many lacking informative alternative descriptions despite legal requirements. Systems like iGraph-Lite and ReVision aim to improve accessibility by providing verbal descriptions and redesigning charts to enhance graphical perception[9]. Furthermore, automatic alt-text systems have been successfully deployed on social networking platforms, demonstrating the potential of AI to generate image descriptions and improve user engagement. Overall, while significant progress has been made in making charts accessible to visually impaired users, ongoing research and development are necessary to address existing limitations and enhance the effectiveness of alternative text solutions[10]

4.4. The Case for and Against Interactive Charts in Email

While the desire to embed interactive charts directly within email communications is understandable, the current capabilities of most email clients present significant limitations. Email clients generally do not support interactive elements like JavaScript, which are often the foundation of web-based interactive charts. This is primarily due to security concerns and the need for a consistent and predictable rendering environment. Embedding fully functional interactive charts directly within the body of an email is therefore not a reliable option for general use.

Attempts to embed complex HTML containing interactive elements, such as those generated by libraries like dygraphs.r, might technically work in some email clients like Thunderbird. However, this approach can lead to inconsistent rendering across different clients, and it might also raise security concerns or trigger warnings, making it an unreliable strategy for broad communication.

A more practical approach to providing some level of enhanced functionality is to embed a static image of the chart within the email and then wrap it with a hyperlink that points to an interactive or scalable version of the chart hosted online. The SVG (Scalable Vector Graphics) format is often a good choice for the linked version, as it allows for zooming and panning in web browsers without loss of quality. This method allows recipients who are interested in exploring the data in more detail to do so by clicking on the chart image, while still providing a readily viewable static representation within the email itself.

5. Conclusion: Selecting the Optimal Solution for Your Email Charting Needs

The landscape of solutions for sending emails with charts is diverse, offering a range of techniques and tools to cater to various needs and technical abilities. The most universally compatible approach remains embedding charts as static images within the email body, leveraging the HTML <img> tag. This method ensures that recipients can view the visual data regardless of their email client or device.

For users seeking to simplify the process, online chart generation APIs like QuickChart.io and Image-Charts provide a convenient way to create chart images on demand by providing chart configurations as URL parameters. These services abstract away the complexities of rendering and return image URLs that can be easily embedded in emails.

The choice of image format (PNG, JPEG, GIF) should be carefully considered based on the chart's characteristics and the desired balance between visual quality and file size. PNG is generally best for detailed charts with text, JPEG for photographs and complex charts where some quality loss is acceptable for smaller file sizes, and GIF for simple graphics and animations with a limited color palette.

Ultimately, the optimal solution depends on the source of the chart data, the desired level of interactivity, the technical expertise available, and the specific requirements for cross-client compatibility and accessibility. By understanding the various techniques and tools available, users can select the most effective approach to enhance their email communications with clear and engaging visual data representations.

REFERENCES:

1. Oppermann, M., & Munzner, T. (2021). VizSnippets: Compressing Visualization Bundles Into Representative Previews for Browsing Visualization Collections. *IEEE Transactions on Visualization and Computer Graphics*. https://doi.org/10.1109/TVCG.2021.3114841
2. Jakobsen, M. R., & Hornbæk, K. (2013). Interactive Visualizations on Large and Small Displays: The Interrelation of Display Size, Information Space, and Scale. *IEEE Transactions on Visualization and Computer Graphics*. https://doi.org/10.1109/TVCG.2013.170
3. Stengel, D., Calori, G. M., & Giannoudis, P. V. (2008). Graphical data presentation. *Injury-International Journal of The Care of The Injured*. https://doi.org/10.1016/J.INJURY.2008.01.050
4. Ajani, K., Lee, E., Xiong, C., Knaflic, C. N., Kemper, W., & Franconeri, S. (2021). Declutter and Focus: Empirically Evaluating Design Guidelines for Effective Data Communication. *IEEE Transactions on Visualization and Computer Graphics*. https://doi.org/10.1109/TVCG.2021.3068337
5. Lu, Q., Zhang, Q., Luo, X., & Fang, F. (2019). *An Email Visualization System Based on Event Analysis*. https://doi.org/10.1007/978-981-15-1377-0_51
6. lcaraz-Martínez, R., & Ribera-Turró, M. (2020). An evaluation of accessibility of Covid-19 statistical charts of governments and health organisations for people with low vision. *Profesional De La Informacion*. https://doi.org/10.3145/EPI.2020.SEP.14
7. Lundgard, A., & Satyanarayan, A. (2021). Accessible Visualization via Natural Language Descriptions: A Four-Level Model of Semantic Content. *IEEE Transactions on Visualization and Computer Graphics*. https://doi.org/10.1109/TVCG.2021.3114770
8. Demir, S. (2008). TAIG: textually accessible information graphics. *Conference on Computers and Accessibility*. https://doi.org/10.1145/1414471.1414555
9. Moraes, P., Sina, G., McCoy, K. F., & Carberry, S. (2014). Evaluating the accessibility of line graphs through textual summaries for visually impaired users. *Conference on Computers and Accessibility*. https://doi.org/10.1145/2661334.2661368
10. Shahira, K. C., & Lijiya, A. (2021). Towards Assisting the Visually Impaired: A Review on Techniques for Decoding the Visual Data From Chart Images. *IEEE Access*. https://doi.org/10.1109/ACCESS.2021.3069205