AdSnap: AI-Powered Ad Banner Generator

Mridul Khanna
Email: khannamridul20@gmail.com

ABSTRACT

In today's digital landscape, businesses heavily rely on digital platforms for marketing. The demand for scalable and consistent ad banners continues to rise. Creating digital advertisements through the traditional approach can be time consuming and often lacks scalability. This project introduces AdSnap: a smart solution focusing on creating visually appealing advertisement banners effortlessly. A publicly available **CGL-Design-v2** dataset that mimics real world complexity of graphic layout designs is used for this study. Leveraging the power of deep learning, we generate slogans tailored to products and creative tone. The posters are generated using modular architecture which performs smart slogan placement, dynamic font color and re-sizing, and contrast color correction to maintain aesthetic balance. Our approach produces high quality production ready ad banners with CTA segment. This study showcases how deep learning and digital media computing can be integrated to streamline creative workflows, which can be easily extendable to future visual marketing initiative.

Index Terms— Advertisements, OpenCV, Large language model, Artificial Intelligence, Digital media computing

1. INTRODUCTION

1.1 Background

According to a report by Forbes [1], 76% of the customers are through social media. This portrays digital marketing as a mandatory skillset for most businesses.

Manually creating digital advertisements requires design expertise, creativity, manual effort and lots of time, making it a bottleneck for small businesses to grow. Marketing teams are often under constant pressure to create digital marketing content in short amount of time [2]. Marketing design teams are expected to create custom advertisement banners catering to high volumes of people over diverse platforms, websites, applications, etc. This challenge highlights the need for automating the process of creating advertisement banners.

Advances in computer vision and artificial intelligence have unlocked techniques for automating visual and textual content in a customized format. Our model leverages the integration of large language models (LLM) and digital media intelligence to create visually appealing ad banners with minimal human input. On providing a tone for the banner (eg Playful, Luxury, Gentle etc), a product name and image, our pipeline generates attention-grabbing slogans, detects ideal placement for it, and balances visual harmony by adjusting color contrast, text resizing and text overlay, thereby producing high quality visual banners with CTA segment.

Several systems have been focusing on automating different aspects of advertisements and content creation, yet most lack a combined approach focusing on modular architecture, scalability, style control, and practical design constraints.

1.2 Related Work

In Automated Banner generation is growing recently with various strategies exploring text generation, style adaptation, layout composition and visual coherence.

AutoPoster [3] proposes a full-fledged advertising poster generation pipeline that transforms a product image and target tone to a deployable banner. The implementation consists of 4 modules: tagline generation using pre trained LLM's, image cleaning to enhance quality and visibility, layout and style generation. Our model draws inspiration from the structure of this model, but we have included dynamic tone variation for slogan generation and rendering layout through bounding box selection.

DiffLayout [4] have proposed a denoising based diffusion framework for generating design poster layouts, given product image and category embeddings as input. By refining noise iteratively, is it able to generate visually balanced formats with layout diversity. This model motivates our choice to use largest bounding box for slogan rendering to maintain visual harmony.

BannerAdency [5] uses multimodal large language models (MLLMs) to generate prompt based banner, given product images, logo and prompts as input. It produces high fidelity editable vector formats like SVG and Figma.   Their model heavily depends on prompt engineering, our model focuses on visual identity elements like text placement and slogan styling.

BrandDiffusion [6] focuses on brand coherence by learning brand's visual identity using brand embeddings and textual inversion with regularization. It learns visual cues from older posters of the brands to maintain brand consistency. They do not focus on user creative control or slogan adaptability.

LayoutTrans [7] proposed diffusion-based layout optimization strategy where the visual elements are predicted token-by-token, thereby allowing control over placement of elements. We also focus on controlling information hierarchy by prioritizing largest bounding box for rendering slogan.

These works demonstrate unique techniques to automate the designing process for advertisements, but none of them combine visual textual harmony, LLM slogan generation, visual contrast optimization, and tone-aware text rendering, our model builds on these ideas to produce a production ready advertisement banner.

1.3 Significance

In the industrial context for the current digital marketing landscape, this project could be a valuable tool to solve the need for scalable and quick advertisement generation tools. Our solution empowers small to mid-size enterprises with a creative and affordable solution for catchy banners.

In the academic context, this project could be a good example to demonstrate how Artificial Intelligence, in particular OpenCV, LLM's and Digital media design could be used in creative fields for content creation.

2. DATASET

We have selected the **Creative Graphic Layout (CGL) Dataset v2**, a dataset provided by hugging face library. It is a publicly available dataset containing graphic layout of posters, with real world complexity.

The dataset consists of 60,000+ records, which includes data about:

- image – A product image (PIL image object)

- text_features – A dictionary with key “pos”, which contains positions of bounding boxes for text, logo, and price
- annotations – COCO annotations with fields like category, box, segmentation, etc.
- Image_id
- File_name
- text_annotations – OCR-related text recognition fields (not used in this project)

2.1 Preprocessing & Filtering

For processing ease, only a subset of this dataset was used for this study. From 62,000+ records, 1000 records were filtered, followed by randomly selecting 50 samples. To the selected sample set, product name and tone for each record was manually added for slogan generation module.

- The text_features column was converted from string to dictionary using ast.literal_eval()
- Images were converted to RGB color format for consistent processing
- Products with at least 1 bounding box annotation was retained. For every sample’s text_features column was parsed and checked to confirm the presence of a non-empty list under the "pos" key. Those samples without any layout information were excluded from further processing.

2.2 Dataset Suitability

After extensive research and experimentation, this dataset was selected for this study. It is well-suited to this project as it contains high quality product images ranging over diverse categories, rich annotations for slogan placement.

3. IMPLEMENTATION

This system is designed to create advertisement banners, on inputting product image, name and tone for the banner. The design of this system is implemented using OpenAI API and various Python libraries such as NumPy, Matplotlib, Pillow etc.

This system consists of various interconnected modules, described below:

3.1 Dataset Processing

The system uses a publicly available dataset - the Creative Graphic Layouts Dataset (CGL-Dataset-v2) from the Hugging Face library

In the dataset, each bounding box is given as a tuple:
(x, y, width, height) ;
where x, y denotes the top-left coordinate, and width × height denote the area where text can be rendered i.e. the bounding box.

Through this we only focused on the images where at least 1 bounding box annotation is present in the text_features["pos"] list.

3.2 Slogan Generation using LLM

We have used OpenAI’s GPT-3.5-Turbo model for generating slogans for product images, given product image and desired tone of the poster.

The model was passed a prompt “You are a world-class advertising copywriter. Give 3 short, punchy, and emotionally appealing ad slogans for a product called ‘X’ with a ‘Y’ tone. Each slogan should be under 10 words and highly brandable. Separate them with line breaks only — no bullet points."

This was followed by scoring the 3 generated slogans and selecting the most appropriate. The scoring rule focused on Medium-length slogans (6–9 words) and use of punctuation (like ! or ?)

Score(s) = s.count("!") + s.count("?") + min(len(s.split()), 10)

The function rewarded concise and emotional slogans. The selected slogan was used in generating the final ad banner.

3.3 Bounding Box Selection

All "pos" records were parsed as a list of bounding boxes for every product image. The largest bounding box by area is selected from the list of annotations, to maximise text visibility.

$$\text{Area} = \text{width} \times \text{height}$$

This ensured the slogan is placed at the most highlighted spot. As the bounding box annotations were already provided, they did not overlap the product core visuals.

3.4 Adaptive Slogan Placement

Python Imaging Library (PIL) handled the Slogan placement. To make the text visually balanced, it is formatted by splitting into two parts at the halfway mark. The first part was placed in bold in the larger font and the second part was placed in normal weight and smaller font below.

The largest font sizes that would fit within the bounding box was calculated by iteratively reducing font size until the text width fit neatly within the box width.

To preserve readability of the text, the text color was decided based on the brightness of the background area inside the bounding box.

The contrast was computed as the weighted brightness using the formula:

$$\mathbf{brightness = R\times0.299 + G\times0.587 + B\times0.114}$$

If brightness > 180 → use black text; else → use white text.

3.5 CTA Block Rendering

A Call-to-Action (CTA) bar was added to each banner at the footer. To maintain consistency of color, the background color was decided based on the bottom 15% of the image. The text color was decided using the same brightness-based logic defined above.

The CTA banner included:

- A rounded CTA button i.e. "Buy Now"
- A price randomly decided e.g., $22.22 AUD)
- A tagline i.e. "Hurry! Limited Time Only"

3.6 Visual Output Display

To display the final outputs, we have randomly displayed some posters side-by-side using Matplotlib. Only valid samples with bounding boxes were visualized.

4. RESULTS

To determine the effectiveness of our model, we applied the strategies to 50 products from the Creative Graphic Layouts (CGL-Dataset-v2) dataset.

4.1 Visual Results

For each product, full banner generation pipeline is implemented.

- Stage 1 – Original: Raw Product Image obtained from the dataset.
- Stage 2 – Slogan Added: The generated slogan is rendered in the largest bounding

box using dynamic font resizing and background-aware contrast detection.

- Stage 3 – Final Poster with CTA: Call-To-Action (CTA) is added to the footer below the image, including a price and tagline.

The pipeline is demonstrated below (Original → Slogan → Final Poster with CTA):

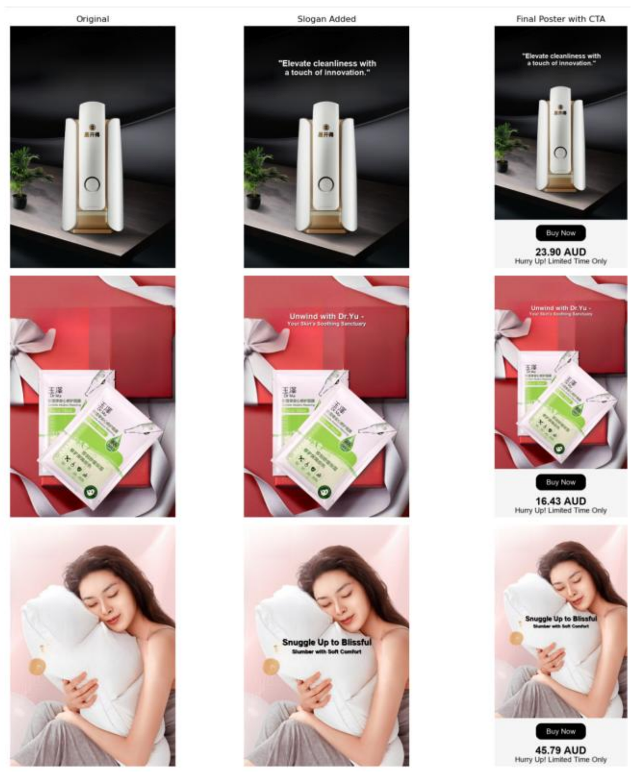
Original
Slogan Added
Final Poster with CTA
"Elevate cleanliness with a touch of innovation."
Buy Now
23.90 AUD
Hurry Up! Limited Time Only
Unwind with Dr.Yu - Your Skin's Soothing Sanctuary
Buy Now
16.43 AUD
Hurry Up! Limited Time Only
Snuggle Up to Blissful Slumber with Soft Comfort
Buy Now
45.79 AUD
Hurry Up! Limited Time Only

Fig 1: Demonstrating AdSnap Pipeline: Original → Slogan → Final Poster with CTA

4.2 Qualitative Evaluation:

We portray some samples of final generated banners in below figure (Fig x). The below images show visually coherent and production ready banners.

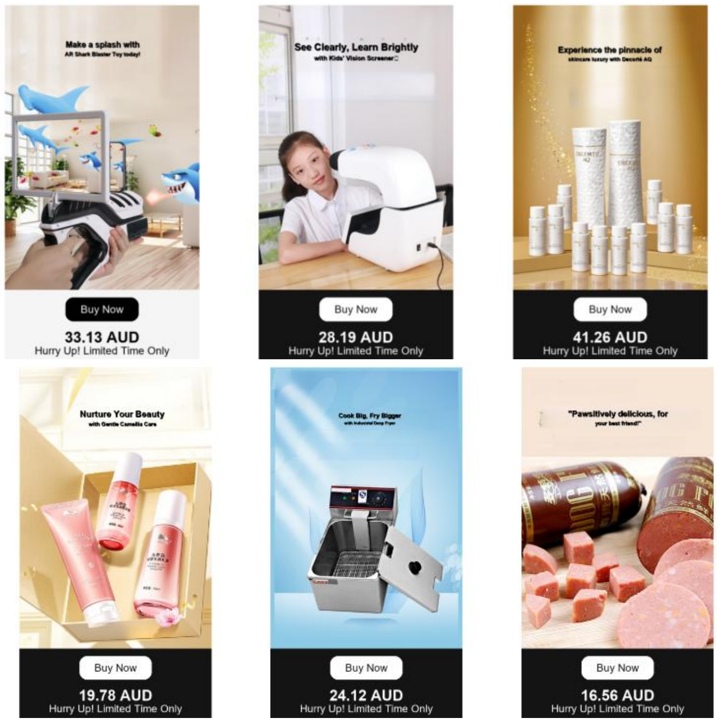
Make a splash with AR Shark Blaster Toy today!
Buy Now
33.13 AUD
Hurry Up! Limited Time Only
See Clearly, Learn Brightly with Kids' Vision Screener
Buy Now
28.19 AUD
Hurry Up! Limited Time Only
Experience the pinnacle of skincare luxury with Decorté AQ
Buy Now
41.26 AUD
Hurry Up! Limited Time Only
Nurture Your Beauty with Gentle Camellia Care
Buy Now
19.78 AUD
Hurry Up! Limited Time Only
Cook Big, Fry Bigger with Industrial Deep Fryer
Buy Now
24.12 AUD
Hurry Up! Limited Time Only
"Pawsitively delicious, for your best friend!"
Buy Now
16.56 AUD
Hurry Up! Limited Time Only

Fig2: Displaying generated Ad banners

The slogans are rendered in the largest bounding box, leading to clean font size and layout.

Using contrast-based text color ensured slogans are readable irrespective of bright or dark background colors.

Tone based slogan generation allowed creative control and helped to align with product's intended marketing feel.

Example tones: "Gentle" for skincare for electronics, and "Premium" for luxury items.

4.3 Runtime Evaluation

To understand the performance efficiency of our model, we measured the execution time for slogan rendering and CTA rendering for 10 samples.

Component	Avg. Time per Sample
Slogan Fitting	0.10 seconds
CTA Rendering	0.16 seconds

Table 1: Run Time evaluation of different components

This shows that the developed model is quick and thereby suitable for batch production of ad banners.

5. DISCUSSION

The proposed model demonstrates a novel system that integrates artificial intelligence and digital media to generate creative and visually appealing ad banners.

5.1 Slogan fitting trade-off

Although our generated banners validate lightweight, production ready banners, there is room for improvement. In real world images, bounding box sizes are inconsistent and often small.

We focused on dynamic font sizing and selecting the largest bounding box. Although our model was successful in rendering slogans, but some of them overflowed, but still visual readability was maintained due to two line formatting and flexible sizing.

stricter control should be performed over llm's over the length of slogans to improve performance.

5.2 Flexibility and real-time potential

The execution time for generating final poster is fast enough for batch production, catalogue generation, and real time deployment for marketing teams or small-scale businesses.

Unlike diffusion-based models which take much longer time, our method focuses on speed and quality.

5.3 Limitations and future improvements:

A) **Layout diversity**

<u>Limitation:</u> only consuming the largest bounding box, though some images had multiple text annotations
<u>Future enhancement</u>: add text fitting multiple boxes

B) **tone-adaptive styling**

<u>Limitation:</u> tone only decides the slogan generated by LLM and not color or style of the banner
<u>Future enhancement:</u> train a tone-to-style color generator

C) Text fitting

<u>Limitation:</u> overflow for longer slogans
<u>Future enhancement:</u> add LLM length control or fine-tuning

D) Evaluation

<u>Limitation:</u> no comparison with human work
<u>Future enhancement:</u> include comparison study

6. Conclusion

In this project we have successfully implemented a modular AI powered advertisement banner generation pipeline, which transforms raw inputs like product image, tone input, product name into visually coherent and lightweight banners.

Our modular architecture provides flexibility and interpretability for the entire pipeline – from slogan generation to dynamic color contrast, font sizing, textbox selection and background color analysis.

Our results show promising outputs, especially where bounding box are not overlapped on the product images and are of reasonable dimensions. However, presence of inconsistent slogan fitting and lack of layout diversity portray potential opportunities for innovation in future.

In future advancements, we plan to integrate:

- Tone style transfer
- Quantitative evaluation pipeline with user feedback to improve quality and provide refinement.
- Eye tracking inspired layout

This model serves as a strong foundation for generating marketing creatives and reinforces the potential of using ai for brand communication and creativity. This can be extended to creating visuals for catalogues, digital marketing campaigns and ecommerce platforms.

REFERENCES

[1] R. Wells, *Social media marketing skills in demand: Worth $1.5 trillion by 2030*, Forbes. [Online]. Available: https://www.forbes.com/sites/rachelwells/2024/01/09/social-media-marketing-skills-in-demand-worth-15-trillion-by-2030/. [Accessed: Mar. 31, 2025].

[2] Digital Marketing Institute, *AI in digital marketing: The ultimate guide*. [Online]. Available: https://digitalmarketinginstitute.com/blog/ai-in-digital-marketing-the-ultimate-guide. [Accessed: Mar. 31, 2025].

[3] J. Lin *et al.*, "AutoPoster: A highly automatic and content-aware design system for advertising poster generation," in *Proc. 31st ACM Int. Conf. Multimedia (MM '23)*, 2023. [Online]. Available: https://doi.org/10.1145/3581783.3611930. [Accessed: Mar. 31, 2025].

[4] H. Zhou, W. Xia, J. Xu, X. Peng, S. Liu, and N. Yu, "LayoutDiffusion: Ad layout generation with denoising diffusion model," *arXiv preprint*, 2023. [Online]. Available: https://arxiv.org/abs/2309.16120. [Accessed: Mar. 31, 2025].

[5] H. Wang, Y. Shimose, and S. Takamatsu, "BannerAgency: Advertising banner design with multimodal LLM agents," *arXiv preprint*, 2025. [Online]. Available: https://arxiv.org/abs/2503.11060. [Accessed: Mar. 31, 2025].

[6] Y. Fong and A. See, "BrandDiffusion: Brand-personalized visual generation using diffusion models," in *Proc. 29th Int. Conf. Intelligent User Interfaces (IUI '24)*, 2024. [Online]. Available: https://doi.org/10.1145/3688867.3690175. [Accessed: Mar. 31, 2025].

[7] Gu, Z., Zhu, J., Zhou, P., Lin, T., Qian, Q., & Lu, H. (2022). LayoutTrans: Unsupervised layout transformation for controllable graphic design. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, 16277–16286. https://doi.org/10.1109/CVPR52688.2022.01583

[8] Creative Graphic Design Lab. (2023). *CGL-Dataset-v2* [Dataset]. Hugging Face. https://huggingface.co/datasets/creative-graphic-design/CGL-Dataset-v2

[9] OpenAI. (n.d.). Models overview. OpenAI. Retrieved May 5, 2025, from https://platform.openai.com/docs/models

[10] Python Software Foundation. (n.d.). *Pillow: Image module*. Pillow Documentation (stable). Retrieved May 8, 2025, from https://pillow.readthedocs.io/en/stable/reference/Image.html

[11] Hunter, J. D., Droettboom, M., Caswell, T. A., Firing, E., & the Matplotlib development team. (n.d.). *Matplotlib: Visualization with Python*. Matplotlib. Retrieved May 6, 2025, from https://matplotlib.org/stable/index.html

[12] Python Software Foundation. (n.d.). *Pillow: ImageDraw module*. Pillow Documentation. Retrieved May 12, 2025, from https://pillow.readthedocs.io/en/stable/reference/ImageDraw.html

[13] Harris, C. R., et al. (2020). *Array programming with NumPy*. Nature, 585(7825), 357–362. https://numpy.org/doc/stable/

[14] Python Software Foundation. (n.d.). *ast — Abstract Syntax Trees*. Python 3 Documentation. Retrieved May 9, 2025, from https://docs.python.org/3/library/ast.html

[15] Python Software Foundation. (n.d.). *time — Time access and conversions*. Python 3 Documentation. Retrieved May 12, 2025, from https://docs.python.org/3/library/time.html

[16] Python Software Foundation. (n.d.). *random — Generate pseudo-random numbers*. Python 3 Documentation. Retrieved May 11, 2025, from https://docs.python.org/3/library/random.html

[17] Python Software Foundation. (n.d.). *colorsys — Conversions between color systems*. Python

3 Documentation. Retrieved May 11, 2025, from https://docs.python.org/3/library/colorsys.html

[18] IPython. (n.d.). *IPython.display — Display rich representations*. Retrieved May 12, 2025, from https://ipython.readthedocs.io/en/stable/api/generated/IPython.display.html