


Saliency Map Prediction Using Vision Transformers with Text Channel Integration

Francesco Uccelli¹ 

Università degli Studi di Milano Statale francesco.uccelli@studenti.unimi.com

Abstract. We present a novel approach for saliency map prediction by integrating text detection into a Vision Transformer (ViT) framework. Our method leverages EasyOCR for text region detection, adding a fourth channel to the input images to enhance the model’s focus on text areas. We fine-tune a ViT model with an adapted patch embedding layer and combine it with a Convolutional Neural Network (CNN) for local feature extraction.

Keywords: Saliency · Visual Transformer · Comicbooks

1 Introduction

This work is submitted for the AI4VA saliency prediction challenge, which focuses on applying computer vision techniques to visual arts, particularly comic art. The challenge involves developing models that predict human visual attention in comic images from the AI4VA dataset.

Our approach addresses challenges of saliency estimation in comics by integrating text detection into the model, considering that text draws attention. We utilize EasyOCR to detect text regions and incorporate them as an additional input channel. Then, we feed this channel to a CNN and fine-tune a Vision Transformer (ViT) for this type of task.

2 Method and training

2.1 Data Preparation

Before training the model, we deal with the data by applying some processes.

- **Image Resizing:** All images are resized to 512×512 pixels to ensure matching to the saliency labels.
- **Normalization:** Images are normalized using mean and standard deviation values of 0.5 for each RGB channel.
- **Data Augmentation:** While augmentation transforms (e.g., random cropping) were included and tested, no performance improvement was seen and it was left out of the final solution.

2.2 Text Detection and Mask Creation

To incorporate textual information, we utilized **EasyOCR** to detect text regions within the full images. The process consists of detecting the text and processing it's bounding boxes. We remove titles (boxes that are bigger than a threshold). And then we construct the masks for the 512×512 . EasyOCR is based on CRAFT: Character-Region Awareness For Text detection [1]

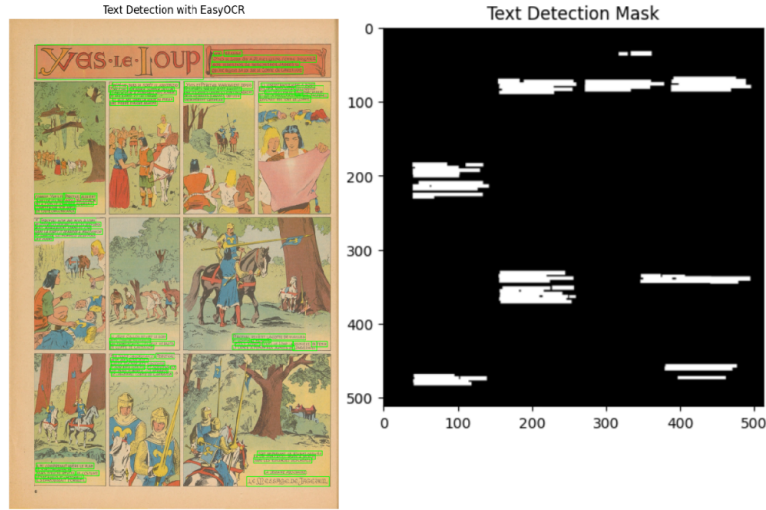


Fig. 1: The image on the left shows the results on the full size image, with bounding boxes in green for different texts. The image on the right shows the mask that we create based on the bounding boxes, now on the resized image, removing titles

2.3 Model Architecture

Our model combines a Vision Transformer (ViT) backbone with a Convolutional Neural Network (CNN) for enhanced feature extraction, specifically tailored to process 4-channel inputs (3 for R,G,B and a fourth with our text mask). Given time limitations, no architecture diagram is presented.

Vision Transformer Backbone

- **Pre-trained Model:** We used the `vit_base_patch16_384` model [2], pre-trained on ImageNet.
- **Patch Embedding Modification:** The original ViT model accepts 3-channel inputs. We modified the patch embedding layer to accept 4-channel inputs by changing the input dimensions of the convolutional layer:

`Conv2d(4, 768, kernel_size = 16, stride = 16)`

- **Transformer Layers:** The ViT processes input patches through standard transformer layers, capturing global context.

CNN Feature Extractor A CNN module extracts local features, which are crucial for capturing fine-grained details in comic images:

- **Architecture Details:** The CNN consists of convolutional layers with ReLU activations and batch normalization, adjusted to accept 4-channel inputs.
- **Feature Maps:** The CNN outputs feature maps that are later fused with ViT features.

Feature Fusion and Upsampling

- **Concatenation of Features:** The output feature maps from the ViT and CNN are concatenated along the channel dimension.
- **Upsampling Layers:** A series of convolutional and transposed convolutional layers are used to upsample the concatenated features back to the original image resolution (512×512):
 - Multiple layers with increasing spatial dimensions to reconstruct the saliency map.
 - Use of ReLU activations and batch normalization to improve convergence.
- **Output Layer:** A final convolutional layer reduces the channel dimension to 1, producing a single-channel saliency map.

2.4 Training and Configuration

We trained the model using the following settings:

- **Gradient Updates:** We froze the ViT backbone’s parameters to focus training on the newly added layers and the CNN feature extractor.
- **Loss Function:** L1Loss gave the best results among the simple loss functions.
- **Optimizer:** Adam optimizer with an initial learning rate of 0.001.
- **Learning Rate Scheduler:** A ReduceLROnPlateau scheduler reduces the learning rate.
- **Device:** Training was conducted on a CUDA-enabled GPU in a [Kaggle notebook](#).
- **GPU:** Training was conducted on Kaggle’s platform using two NVIDIA Tesla T4 GPUs (GPU T4 x2).

2.5 Ensemble and inference

During training, we saw a clear trade-off for AUC and KLD. The model overfit slightly to increase AUC but KLD decreased as epochs incremented. To overcome this, we averaged the predictions of two models: One with good AUC (result of more epochs during training) and one with good KLD (results of less epochs during training). Applying Gaussian smoothing to the predictions also showed minor improvements.

3 Evaluation Results

3.1 Model Development and Performance

We began our experimentation with a simple Convolutional Neural Network (CNN) model trained for 10 epochs. This model yielded moderate results, demonstrating the capability to capture basic features in the images. Building on this, we implemented a Vision Transformer (ViT) model, which showed slightly better performance. We added the OCR-generated text channel to the ViT model, resulting in further improvements. Next, we combined the ViT model with the CNN feature extractor and incorporated the OCR channel, achieving significant gains across all evaluation metrics. The evaluation metrics for each model are summarized in Table 1.

Table 1: Model Evaluation Metrics

Model	AUC	SIM	CC	KLD
CNN (10 epochs)	0.79	0.58	0.26	0.59
ViT	0.80	0.58	0.24	0.56
ViT + OCR	0.81	0.59	0.32	0.54
ViT + CNN + OCR	0.86	0.68	0.62	0.32
Ensemble	0.89	0.73	0.76	0.42

Due to the necessity of balancing all evaluation metrics, qualitative analysis of the predictions was crucial. By closely inspecting the saliency maps produced by each model iteration, we identified strengths and weaknesses not immediately apparent from the quantitative metrics alone. This informed our decision-making process in model development and ensembling.

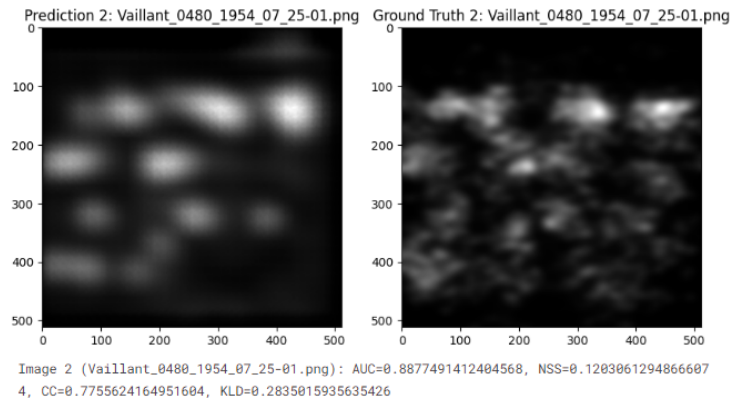


Fig. 2: Example saliency map predictions showing results with validation metrics.

4 Conclusion and Future Work

We introduced a method that integrates text detection into saliency prediction models using a modified Vision Transformer architecture. The inclusion of textual information as an additional channel allows the model to better predict regions that attract human attention, particularly in the context of comic art where text plays a significant role. Through iterative improvements and model ensembling, we achieved significant gains across evaluation metrics.

4.1 Limitations

Despite the promising results, our approach has several limitations:

- **Training Data Size:** The dataset size was relatively small, which may affect the model’s ability to generalize to unseen data.
- **Generalization to Other Styles:** While the model performed well on the provided comic styles, its effectiveness on other artistic styles or real-world images remains to be evaluated.

4.2 Future Work

For future research, we propose the following directions. This could lead to significant improvements, while maintaining the workload in a relatively easy way.

- **Quantitative Evaluation:** Conduct extensive quantitative evaluations using metrics like AUC, CC, SIM, and KLD on larger and more diverse datasets to better assess the model’s performance.
- **Model Optimization:** Optimize the model architecture to reduce computational requirements, specifically aiming to prevent overfitting by adding skip connections or drop out layers.
- **Image resolution:** Train a model to map resolutions of different sizes like the real comics to the saliency maps in 512×512 so as to capture more information
- **Additional data** Use additional data to better the context for the model, find more fitting pretrained weights.

References

1. Baek, Y., Lee, B., Han, D., Yun, S., Lee, H.: Character region awareness for text detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 9365–9374 (2019) [2](#)
2. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale (2021), <https://arxiv.org/abs/2010.11929> [2](#)