**Smelling the Game: Multi-Label and Proportional Scent Prediction from Visual Scenes**

Haoran Dong, Yingyi Zhu, Yuteng Chen

**Github:** <https://github.com/QIQIZHANG852/CSCI1470_Final_Project>

**Introduction**

In the early days of gaming, graphics were simple — often limited to just 8-bit pixel art, as seen in classics like Super Mario. Today, thanks to advancements in AI-generated graphics and cutting-edge rendering technologies, video games have achieved breathtaking levels of realism and detail, delivering immersive visual experiences that were once unimaginable. This dramatic leap in visual fidelity led us, a group of avid gamers, to ask an important question: **What other senses could be enhanced to make games even more immersive?**

We noticed that while modern games are more immersive and visually diverse than ever, much of this immersion is focused on sight and sound, leaving other sensory cues—such as scent—largely underexplored. One natural next step is the sense of smell — an often overlooked but powerful component of human experience. Our project aims to address this gap by building a model that learns to associate “scent profiles” with specific game scenes. In other words, we want the model to look at a screenshot from a game and infer which scents might be present (e.g., “muddy,” “fresh,” “chemical,” “bloody”), and at what relative intensities.

We explore how to bring scents into gaming by leveraging deep learning (DL) and computer vision (CV) models. This model forms the core of a broader vision: to enable real-time scent generation in games through an intelligent scent-emitting device. By accurately predicting scent profiles from visuals, the model provides the essential bridge between the digital environment and the physical sensory experience, ultimately aiming to enhance gameplay immersion through smell. We view this primarily as a multi-label classification task, because the output is a normalized set of intensities across multiple scent categories. By predicting these scent distributions, we hope to lay a foundation for enhancing player immersion through more nuanced in-game feedback—potentially influencing how games could respond dynamically to a player’s sensory preferences.

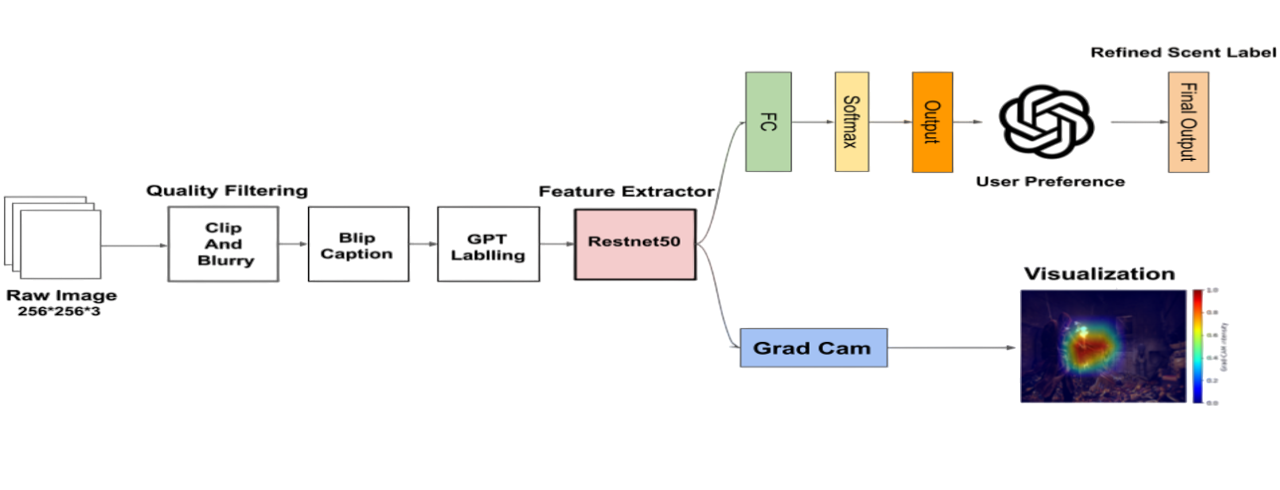
**Preprocessing / Data**

Our primary dataset comes from this Hugging Face repository, which originally contained 52,726 images from various video games. Each entry includes a screenshot and textual metadata (e.g., game name, captions, etc.). These images are diverse in content, spanning open-world titles, stealth games, fantasy RPGs, and more. Because not all images are equally informative (some are blurry or lack clear context), we perform a filtering step to remove low-quality screenshots. Afterward, we generate or refine captions and “scent” labels for each remaining image. The final dataset used for model training will include roughly 2,678 images, each with a detailed caption and a normalized scent distribution label (e.g., {“muddy”: 0.4, “bloody”: 0.3, …}). This labeled data then drives our multi-label classification experiments. Minimal additional preprocessing is required beyond our quality checks and label generation, so the dataset is otherwise used as-is, leveraging the diversity of its original game images to ensure broad coverage of potential “scents.”

**Model Architecture**

Our implementation includes a visual quality filtering pipeline and a ResNet-based scent prediction model, followed by an optional GPT-based user preference module for personalized scent refinement.

***Pipeline***



**Fig 1.** Our work pipeline includes dataset preparation and model architecture.

Our model architecture consists of a ResNet-50 backbone that processes each 224×224 RGB image to extract visual features. These features were then passed through two fully connected layers and we reduced the dimensionality from 256 to 21, followed by a softmax layer that outputs a probability distribution over the predefined scent categories. Additionally, we include an optional user preference tuning module: by using GPT-3.5, users can input natural-language feedback (e.g., “avoid pungent smells”), allowing the system to refine the predicted scent profile to better match individual preferences.

To train our model, we split the dataset into 80% training and 20% validation sets, using a fixed random seed to ensure reproducibility. The model was optimized using the Adam optimizer with a learning rate of 1e-5 and a batch size of 32, and trained over 20 epochs. We used a combination of KL divergence loss on soft labels and an L1 output penalty to encourage sparse and decisive predictions.

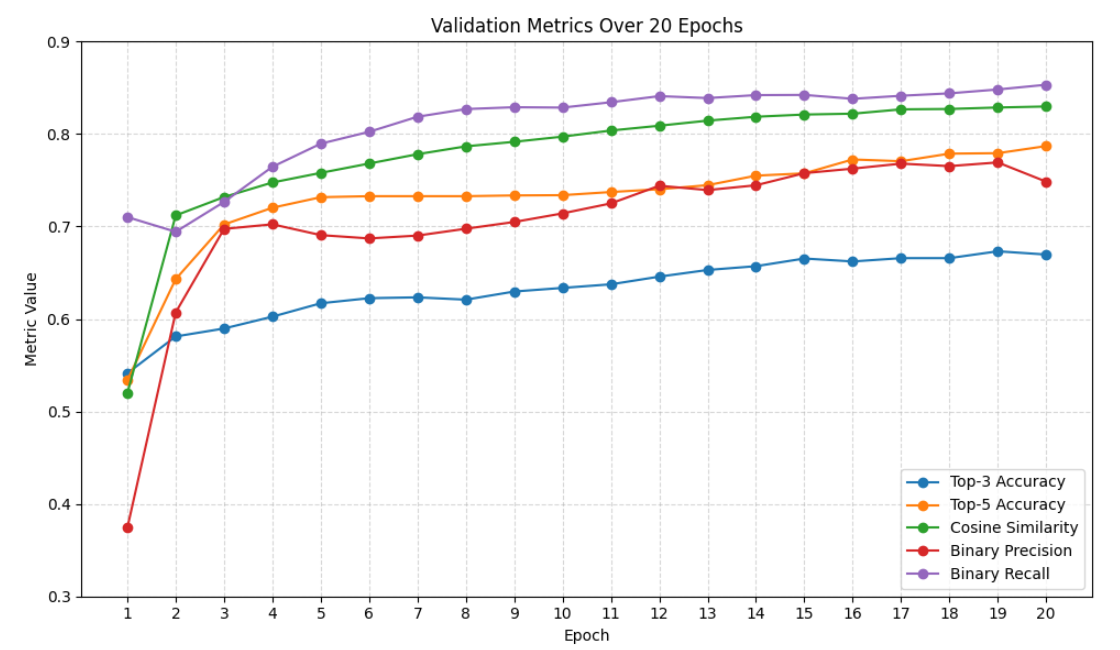
Meanwhile, as for model interpretability, we applied Grad-CAM, an interpretability technique designed to shed light on the “reasoning” inside a convolutional neural network. Rather than treating the network as a black box, Grad-CAM produces a coarse heat-map over the input image indicating which regions most strongly influenced the model’s decision for a particular class label—such as “muddy” or “bloody” in the scent-prediction task. By visualizing these attention regions, Grad-CAM helps verify that the model is focusing on semantically meaningful cues (e.g., muddy puddles for “muddy” scents or glowing embers for “burnt”). This spatial interpretability is invaluable both for debugging—catching cases where the model might be spurious or overfitting—and for building user trust, since one can show exactly why the model predicts a particular scent profile from a game screenshot.

***Hyperparameters***

| **Batch\_size** | **32** |
| --- | --- |
| **Epoch** | **20** |
| **Learning\_rate** | **1e-5(Adam)** |
| **Activation** | **Relu** |

**Results**

***Evaluation Metrics***

****

**Fig 2. Evaluation metrics over 20 epochs**

The model is assessed using several complementary evaluation metrics suited for multi-label, multi-output prediction tasks:

* **Top-k Coverage** evaluates whether the true labels are captured within the model's top-k predicted scents. For example, Top-3 and Top-5 Accuracy track the fraction of cases where at least one of the true scents appears among the top 3 or 5 predictions. This metric is particularly useful when exact ranking matters more than precise probability estimates.
* **Binary Precision & Recall** assess the model’s ability to predict the presence or absence of each scent, based on a threshold (e.g., 0.5). Precision measures how many of the predicted positives are actually correct, while recall measures how many of the actual positives were successfully predicted. These are crucial for evaluating performance when dealing with sparse or imbalanced labels.
* **Mean Absolute Error (MAE)** quantifies the average absolute difference between predicted and actual scent proportions. Unlike binary metrics, MAE captures how close the predicted scent intensity is to the ground truth, which is valuable when predictions are continuous or proportional rather than binary.
* **Cosine Similarity** measures how aligned the overall predicted scent vector is with the ground-truth vector, treating each sample as a high-dimensional direction. A high cosine similarity indicates that the relative distribution of predicted scents closely matches the true distribution, even if the absolute values differ.

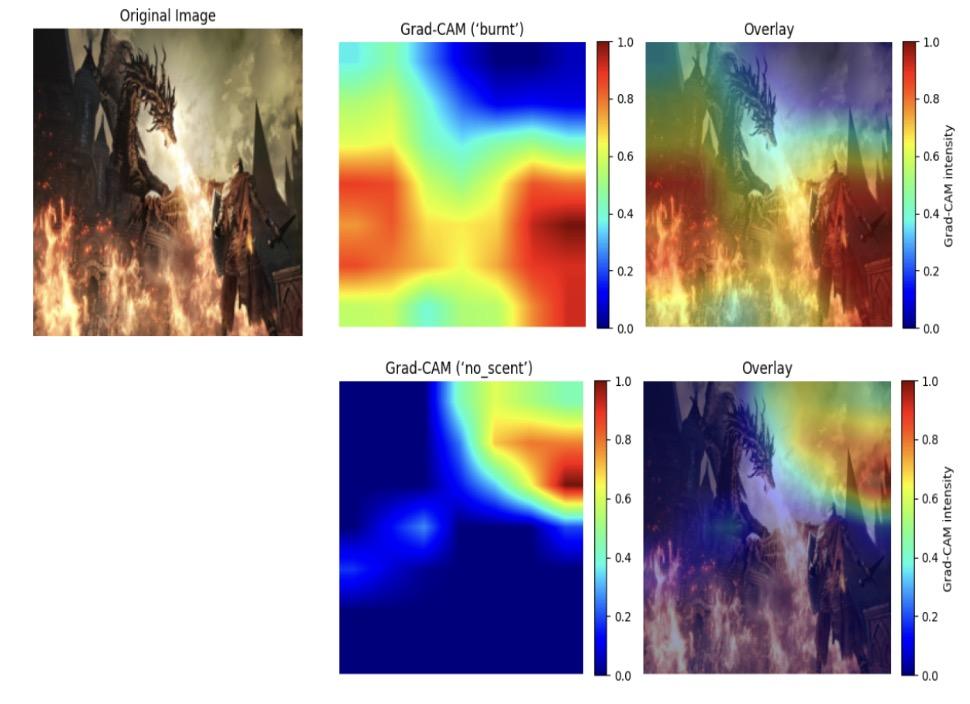
The line plot shows how each evaluation metric evolves over 20 epochs of model training. Both Top-3 Accuracy (blue) and Top-5 Accuracy (orange) increase sharply during the first few epochs and then plateau, suggesting that the model learns to rank the most relevant scents early on. Top-5 consistently outperforms Top-3 due to the broader prediction window.

Cosine Similarity (green) improves steadily and stabilizes around 0.83, reflecting a strong alignment between predicted and true scent distributions. This suggests that the model captures the overall scent profile effectively.

Binary Precision (purple) starts relatively high and continues to improve slightly before leveling off close to 0.88, indicating that most predicted scent presences are correct. Binary Recall (red) shows a more modest improvement and levels off around 0.75, implying that while the model is accurate when it predicts a scent, it misses some of the true ones—a common trade-off in multi-label settings.

Overall, the metrics converge after around 8–10 epochs, indicating the model reaches a performance plateau relatively early. This trend can inform early stopping or model checkpointing decisions during training.

***Grad-CAM Visualization***

******

**Fig 3.** Heatmaps generated by Grad-Cam. Demonstrate where the model looks when predicting different scents.

The top row of the figure presents the Grad-CAM analysis for the “burnt” scent on a single game screenshot. The original image (left) depicts a dragon breathing fire over a village. In the raw Grad-CAM map (center), warmer colors (yellows and reds) correspond to regions that most strongly activate the “burnt” output neuron. These high-intensity areas coincide precisely with the blazing flames and charred rooftops in the foreground. When overlaid onto the RGB image (right), this heat-map confirms that the model’s “burnt” prediction is driven by the fire-laden portions of the scene, indicating that the network has learned to associate visual cues of combustion with the corresponding scent label.

By contrast, the bottom row illustrates the Grad-CAM results for the “no\_scent” label on the same screenshot. Here, the raw heat-map (center) highlights cooler, blue-shaded regions—areas contributing positively to the “no\_scent” score—located in the upper-right background where the sky is clear and devoid of odorous elements. In the overlay (right), these blank regions are emphasized, demonstrating that the model correctly identifies unscented portions of the image when emitting a “no\_scent” prediction. Together, these visualizations provide formal evidence that the network’s scent predictions are grounded in semantically meaningful image regions, thereby enhancing interpretability and trust in the model’s decision process.

*User Preference Adjustment*

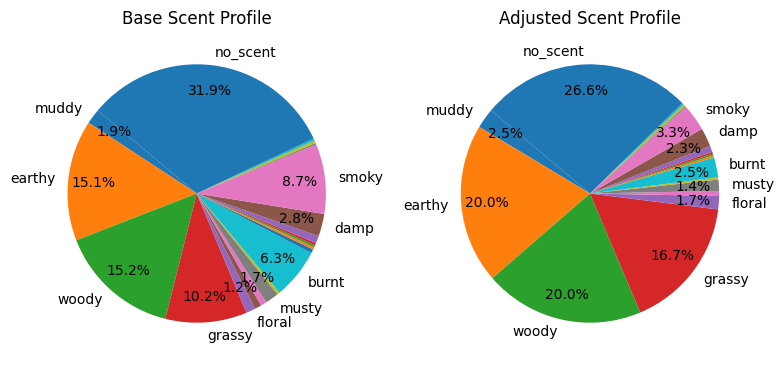
**

Fig 4. Scent profile before and after the adjustment given user input “avoid pungent smells; prefer more natural ones”.

In the Base Scent Profile, the model originally allocated 15.2 % to “woody,” 15.1 % to “earthy,” and 10.2 % to “grassy,” alongside more pungent notes of 8.7 % for “smoky” and 6.3 % for “burnt.” After applying the user preference to “avoid pungent smells; prefer more natural ones,” the Adjusted Scent Profile shifts strikingly: both “woody” and “earthy” rise to 20.0 %, while “grassy” climbs to 16.7 %, reinforcing the desired natural character. Conversely, the sharper olfactory cues are substantially down-weighted—“smoky” declines from 8.7 % to 3.3 % and “burnt” from 6.3 % to 2.5 %. This reallocation demonstrates the adjustment mechanism’s ability to de-emphasize more acrid scent components in favor of the warmer, earth-derived aromas the user specified.

To ensure that the scent distribution remains faithful to the visual scene, the GPT prompt is explicitly crafted to make only moderate adjustments. In practice, this means asking the model to nudge the proportions away from harsh olfactory cues without completely overriding the base prediction. Should a more conservative tuning be desired or a more aggressive personalization, the same prompt framework can be modified accordingly. By varying prompt language (e.g., “slightly increase” versus “strongly prioritize”), one can control the degree of adjustment while maintaining semantic alignment with the image.

**Challenges**

The most challenging part of the project has been constructing a labelled dataset that captures a wide range of scent associations for game environments. Since no public dataset exists for this purpose, we had to collect over 50,000 open-sourced game screenshots from hugging face and develop an automated pipeline using BLIP for image captioning and GPT-3.5 for scent distribution labeling. And Ensuring consistency and interpretability across soft-label distributions required significant prompt engineering and verification.

Another major challenge was the quality of the training data. Although we tried to filter out the original dataset to get some high-quality images, many of them still lacked sufficient clarity or contextual richness, which likely introduced noise and reduced model reliability. This limitation not only affected prediction accuracy but also made it harder for the model to learn fine-grained distinctions between similar scent categories.

**Discussion & Future Work**

There were many limitations in our implementation, especially about our training dataset. Since no existing dataset directly links visual game scenes to scent labels, we had to rely on some open-source image collections and generate our own labels. This led to several challenges. First, due to the inconsistent quality of the raw dataset, we had to implement a filtering pipeline. However, even after filtering, many images lacked sufficient visual clarity, which may have introduced noise and reduced the model’s ability to make accurate predictions. Second, our ground-truth scent labels were generated using GPT-3.5 in a zero-shot setup based on image captions. While this approach enabled scalable annotation, it also introduced potential issues such as subjective interpretations, lack of real-world olfactory grounding, and inconsistency in label distribution—particularly for subtle or uncommon scent categories.

To address these limitations, our future work will focus on improving both data quality and model design. First, we plan to manually annotate a subset of images to establish a more reliable ground-truth baseline, especially for nuanced or ambiguous scent categories. Additionally, we aim to build a new dataset consisting of higher-resolution, semantically rich game scenes to reduce noise and improve model performance. On the modeling side, we intend to replace the fixed ResNet backbone with more advanced architectures which can better capture global context and improve multi-label separation. Finally, we will continue tuning hyperparameters in our ScentPredictors model to achieve more precise and stable scent predictions across diverse visual scenes.

**Reflection**

Our project turned out well, and we met all three of our goals. First, we built and trained a CNN that can look at a single game screenshot and predict a mix of scents, fulfilling our base goal. Then we added Grad-CAM heat-maps so that we can see exactly which parts of each image led to a given scent prediction, which was our target goal. Finally, we wrapped a small language model around the outputs so that players can say things like “avoid pungent smells; prefer natural ones,” and the scent mix adjusts accordingly, which was our stretch goal.

Overall, the model behaves as we expected. For example, when it predicts “woody,” the heat-map lights up on trees or wooden structures; when it predicts “burnt,” it focuses on fire and smoke. The adjusted profiles also shift in the right direction if the user asks to dial back harsher smells and boost earthy or grassy notes. The numbers bear this out, too—our cosine similarity, top-k coverage, and precision/recall metrics all improved compared to a random guess, and the Grad-CAM visuals match our intuition about where true scent cues live in the scene.

If we had more time, the main thing we’d tackle is the dataset. Even after filtering out low-quality screenshots and using BLIP to auto-generate captions, a lot of images still aren’t a perfect fit for predicting scent. BLIP often misses small but important details—like a smoldering campfire in the corner—so our labels end up noisy. Spending more effort on either fine-tuning BLIP or having a quick human check for each caption would tighten up our ground truth and almost certainly boost all our numbers.

The biggest takeaway is how powerful off-the-shelf vision models can be. With only minor tweaks, they turned into reliable scent predictors and provided meaningful Grad-CAM maps. Adding a lightweight language-model layer for user preferences was almost plug-and-play, and it worked smoothly. In the end, this project showed us that by combining a strong CNN backbone with a simple LLM prompt, we can open up a whole new dimension—smell—in gaming in a surprisingly straightforward way.