For SmolVLM-256M-Instruct:

- Processed input size = 512×512

- Channels = 3 (RGB)
  So input tensor = [1, 3, 512, 512]

- Vision Encoder Patch size = 16×16
  So, the image is divided into patches of size 16×16

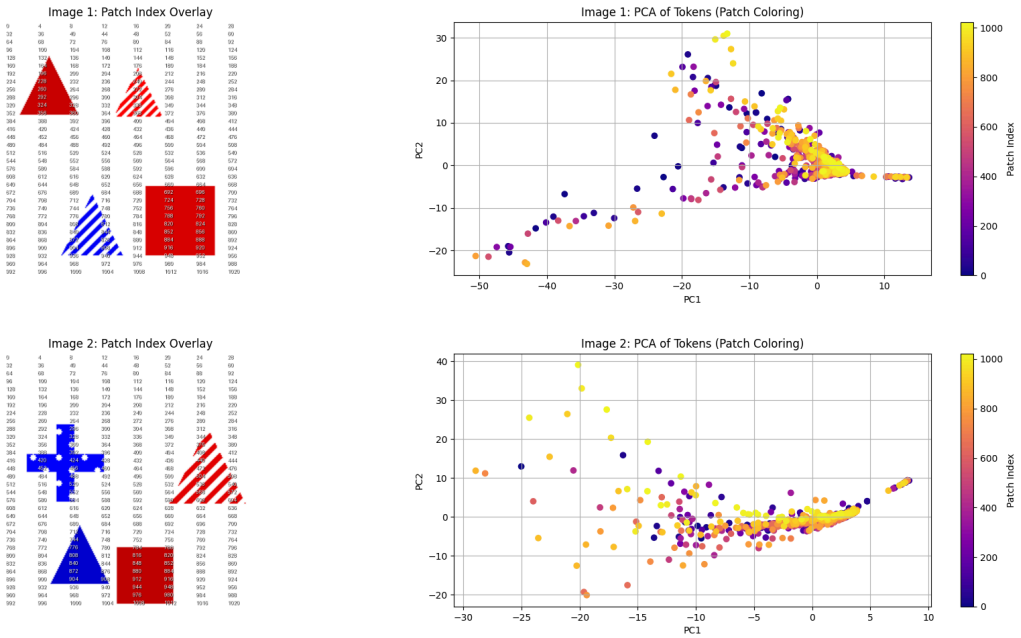
- Number of patches per dimension: 512/16 = 32
  So total number of patches = 32×32=1024 patches

- Each 16×16 patch contains 3 channels (RGB), so each patch has:
  16×16×3=768 values (raw pixel vector)

- This 768-dim vector is then passed through a linear projection layer Linear(in_features=768, out_features=768). Hence we get the hidden representation also of size 768.

- **The result:**
  For each of the 1024 patches, we get a 768-dimensional vector (feature embedding).
  Hence the shape of the output tokens for 1 image is (1024,768)

  Note: The input size, patch size and the hidden size differ from model to model. For example in the case of SmolVLM-Instruct (I.e 1.7B Model), the input size, patch size and the hidden size is 384, 14, 1152 respectively.



The raw tokens are extracted using the visual encoder of the SmolVLM-256M-Instruct.

FIG 1: Original image (512x512)

Each number on the image represents the index for every 4th patch. This will help to correspond the patches to the respective points on the PCA scatterplot.

FIG 2: PCA with Patch Index Coloring

- Each point is a visual token representing a patch from the image. There are total 1024 tokens.
- PCA has reduced their 768-dim token vector to 2D for visualization of each of the image.
- Points close together likely encode similar visual content.
- Each token is colored by its patch index (i.e., spatial position in the image grid of 32x32, from token 0 to token 1024).
- Color bar tells you which part of the image each patch came from: lower indices (e.g. top-left) are be violet, higher indices (e.g. bottom-right) more yellow.

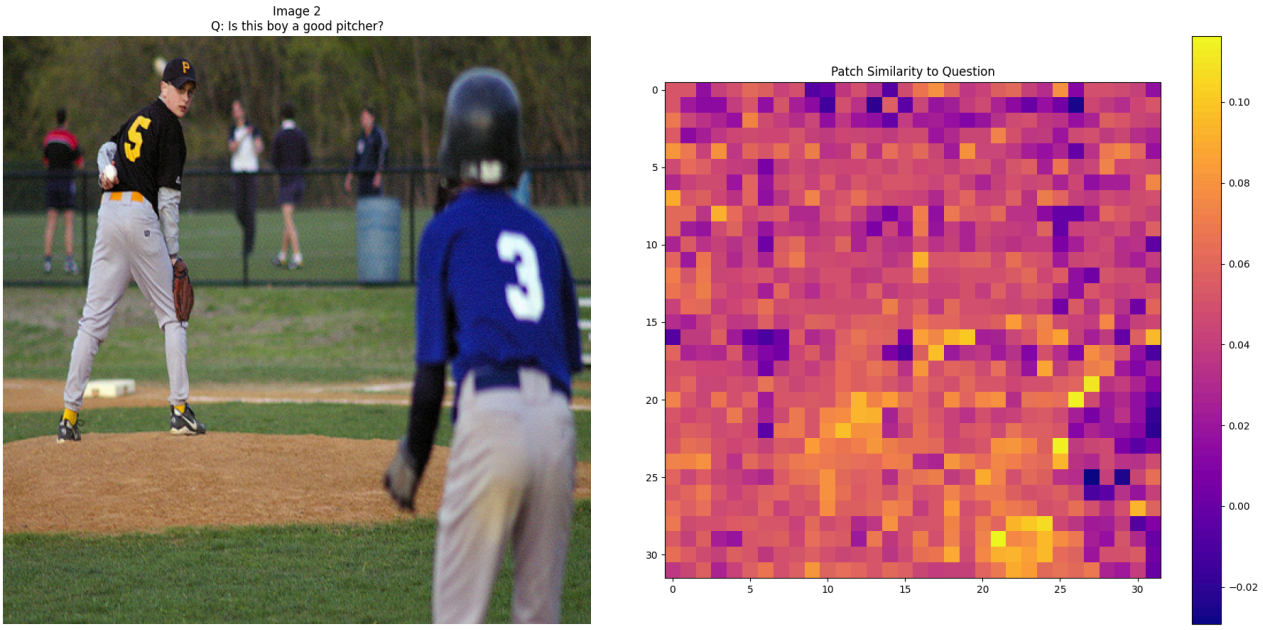
Interpretation:

- Dense clusters = patches with similar textures/semantics. (likely background or less informative ones)
- Spread-out areas = more unique or informative patches. (e.g. colored shapes, patterns, edges)

Observation in PCA (Patch Index Colored)	**What it Suggests**
Smooth gradient (e.g. left -> right color flow):	Encoder respects spatial continuity (e.g., patch 0 -> 1 -> 2 vary gradually)
Dense clusters of same color:	Patches have similar embeddings => possibly flat background
Widely scattered patch indices:	High semantic variation (e.g., patches capturing edges, shapes)
Noisy or random colors:	Poor spatial locality => encoder not capturing position-aware info

Merve-vqav dataset: Cosine similarity between each image patch and the question embedding from SmolVLM-256M-Instruct

Image 2



- Each square (patch) in the heatmap corresponds to a 16×16 region of the image, and the color intensity of that patch indicates how semantically relevant that patch is to the question, according to the model's vision and language representations.

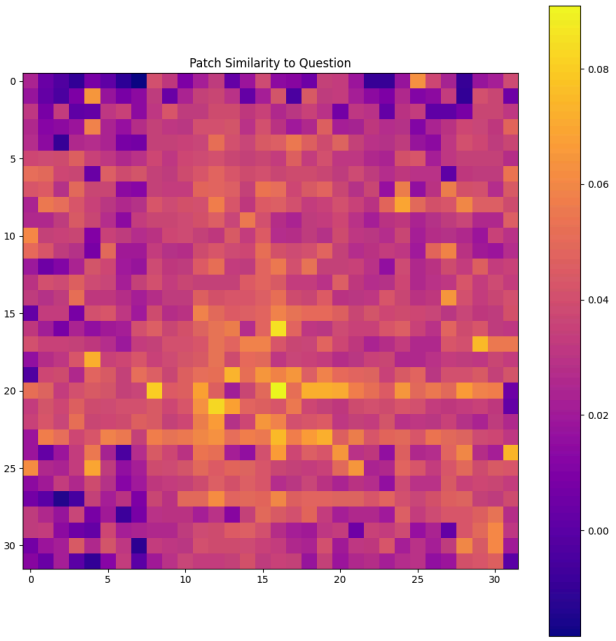
  - Bright (high similarity): The model thinks this patch contains visual content that's relevant to the question.
  - Dark (low similarity): The model thinks this patch is less relevant or background.

- This helps us understand what parts of the image the model "attends" to when answering a question. We can debug cases where the model fails, maybe it's focusing on irrelevant regions.
- Example: In the above image, the bottom part of the heatmap seems to be brighter indicating that the model is giving more attention to the pitch to answer the question.

- Similarly in the image below, certain patches in the center are brighter, indicating the model is rightly focussing on the person's wetsuit.

Image 3


Image 3
Q: What is the person wearing?


Patch Similarity to Question
0
5
10
15
20
25
30
0.08
0.06
0.04
0.02
0.00