DIP PROJECT

SKETCH BASED IMAGE RETRIEVAL SYSTEM

DEBAYAN DAS (201301131)
AMEYA PRABHU (201402004)

WHY SKETCH BASED SEARCH?

ONE PICTURE IS WORTH THOUSAND WORDS. IMAGES PRESENT AN INTUITIVE MEDIUM OF REPRESENTATION. SKETCHES ARE EASIER TO DRAW. ALSO, VERY BIG CORPUS AVAILABLE ON THE INTERNET (GOOGLE IMAGES, PICASA, FLICKR, ETC).

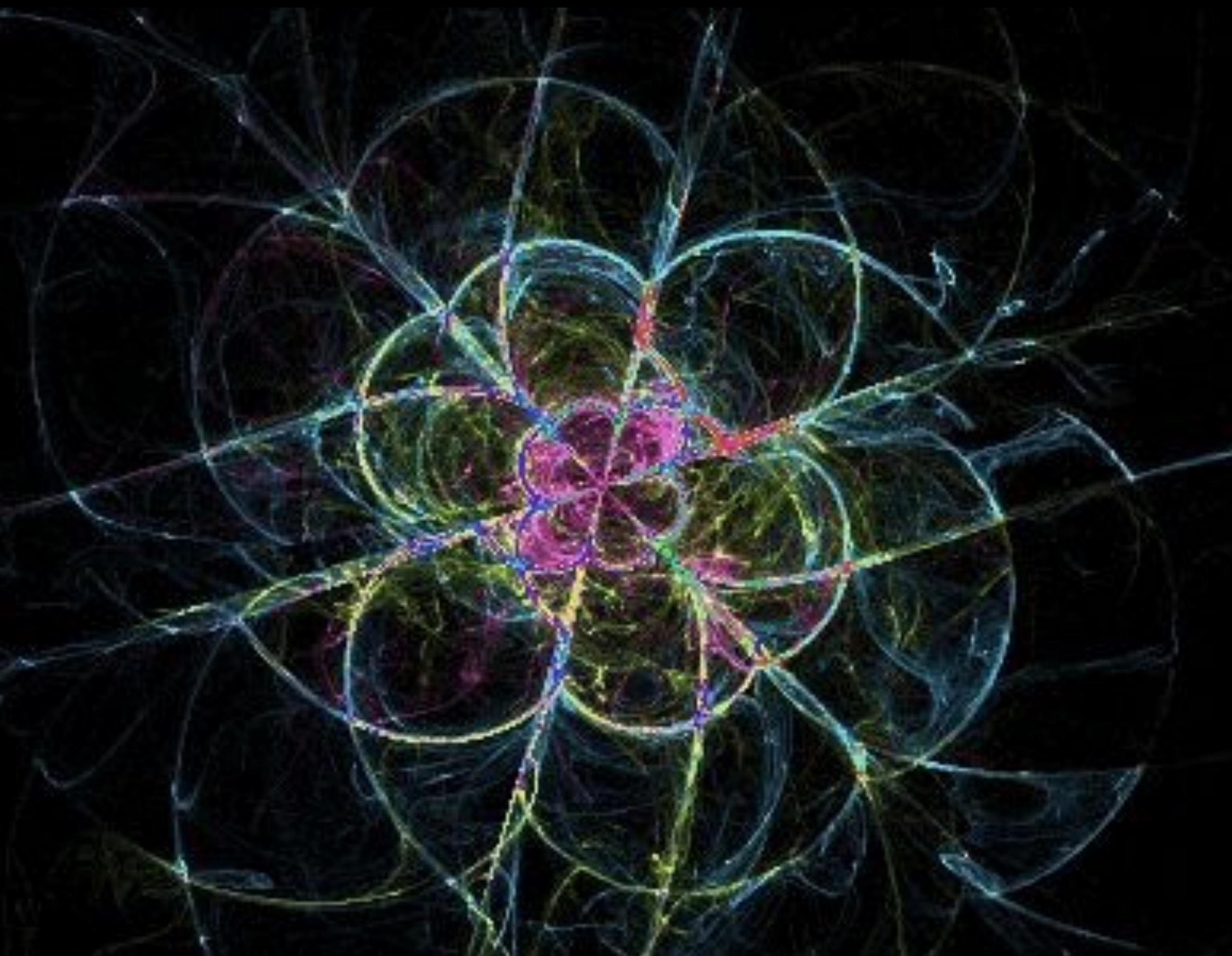
THE PROBLEM

TRADITIONALLY, SEARCH HAS BEEN IN TERMS OF TEXTUAL KEYWORDS. BUT TEXTUAL TAGS ARE TOO SIMPLE TO CONCISELY REPRESENT THE COMPLEX INFO THAT AN IMAGE HOLDS.

IMAGE MATCHING CAN HELP. BUT CANNOT USE SIMPLE COLOR-BASED IMAGE MATCHING (AS SKETCH LACKS ANY SORT OF COLOR INFO). ALSO, DRAWING FINE DETAILS IN A SKETCH IS DIFFICULT (NOT EVERYONE IS AN ARTIST). HENCE, SKETCHES ARE SUITABLE FOR COARSE MATCHING.

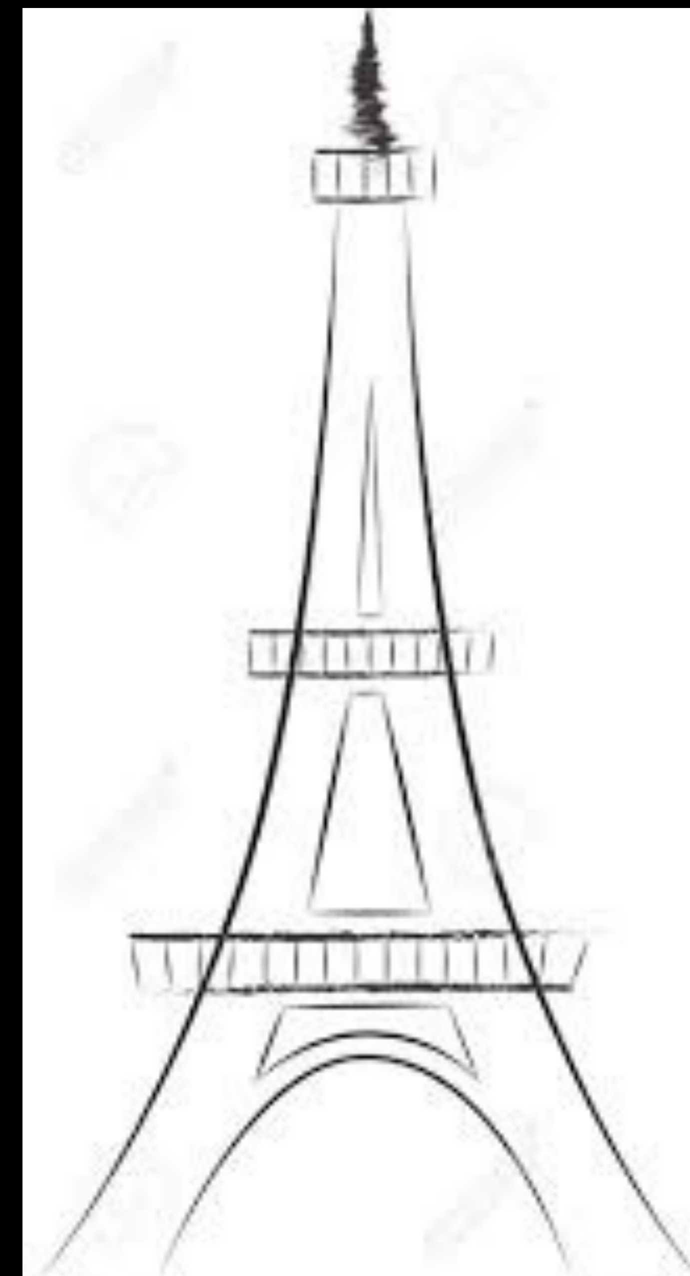
POSITION OF SEARCH OBJECT IN SKETCH AND REAL IMAGE CAN BE DIFFERENT. ALSO, COMPLEX BACKGROUND OBJECTS AND INTERIOR DETAILS OF OBJECTS CAN INTERFERE IN OUR SEARCH.

THE PROBLEM



DESCRIBING SUCH COMPLEX IMAGES WITH MEANINGFUL TEXTUAL TAGS IS DAUNTING AND SOMETIMES IMPOSSIBLE.

OUR SOLUTION



WE NEED TO BRIDGE THE GAP BETWEEN THE RGB COLOR SPACE OF IMAGES IN DATABASE AND THE BINARY COLORSPACE OF QUERY SKETCH.

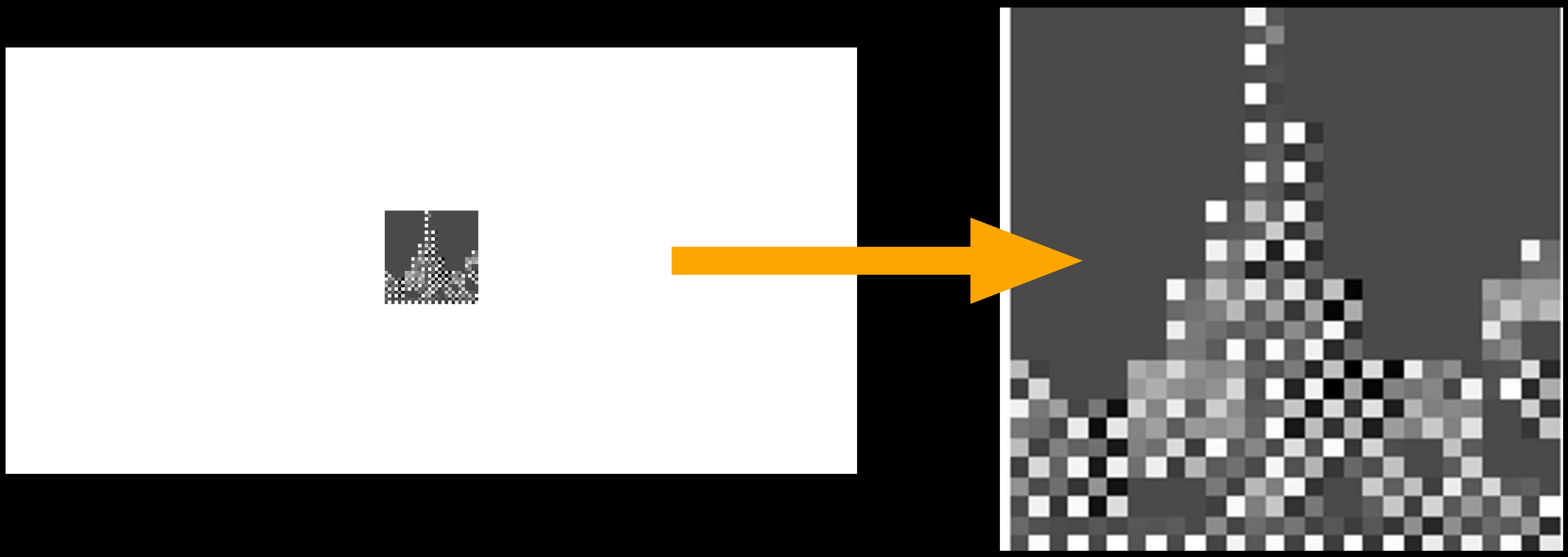
OUR SOLUTION

USE GRADIENT IMAGE. THRESHOLDED GRADIENT OF AN IMAGE CAPTURES DIRECTION OF PRINCIPAL STROKES AND HENCE GIVES IDEA ABOUT THE STRUCTURE OF AN OBJECT DRAWN IN THE SKETCH

INITIAL APPROACH

WE USE A DESCRIPTOR CALLED THE **STRUCTURE TENSOR DESCRIPTOR**.

WE DIVIDE IMAGE INTO 16X16 SIZED BLOCKS AND COMPUTE STRUCTURE TENSOR FOR EACH BLOCK. THESE STRUCTURE TENSORS TOGETHER FORM THE FEATURE VECTOR/DESCRIPTOR FOR THE ENTIRE IMAGE



OUR APPROACH

TO COMPUTE THE STRUCTURE TENSOR, WE FIND GRADIENT OVER ENTIRE IMAGE AND ITERATE OVER ALL BLOCKS. FOR EACH BLOCK, WE COMPUTE THE TENSOR MATRIX. THIS MATRIX ENCODES THE DIRECTION OF THE UNIT VECTOR THAT IS AS PARALLEL AS POSSIBLE TO MAJORITY OF THE GRADIENTS AT THE PIXELS IN THE BLOCK.

Let $\mathbf{x}$ be a unit vector, which we want to define such that it represents the main direction in cell C_{ij}. As $\mathbf{x}^\top \mathbf{g}_{uv}$ attains a maximum if $\mathbf{x} \parallel \mathbf{g}_{uv}$ we pose the definition of $\mathbf{x}$ as the following optimization

$$\mathbf{x} = \arg\max_{\|\mathbf{x}\|=1} \sum_{(u,v)\in C_{ij}} \left(\mathbf{x}^\top \mathbf{g}_{uv}\right)^2 . \tag{6}$$

TO SHOW THAT THE TENSOR MATRIX ENCODES STRUCTURE INFO, WE CAN SHOW THAT THE UNIT VECTOR WE NEED IS AN EIGEN VECTOR OF IT.

OUR APPROACH

$$\sum_{(u,v)\in C_{ij}} \left(\mathbf{x}^\top \mathbf{g}_{uv}\right)^2 = \sum_{(u,v)\in C_{ij}} \mathbf{x}^\top \mathbf{g}_{uv}\mathbf{g}_{uv}^\top \mathbf{x} =$$
$$\mathbf{x}^\top \left(\sum_{(u,v)\in C_{ij}} \mathbf{g}_{uv}\mathbf{g}_{uv}^\top\right) \mathbf{x} = \mathbf{x}^\top G_{ij}\mathbf{x}$$

$$2G_{ij}\mathbf{x} + 2\lambda\mathbf{x} = 0$$

HERE 'G' IS THE TENSOR MATRIX AND IT IS COMPUTED FOR EACH CELL.

In order to detect similarly oriented image edges independently of the magnitude of the edges, we store the structure tensor normalized by its Frobenius norm:

$$T_{ij} = \frac{G_{ij}}{\|G_{ij}\|_F} \tag{9}$$

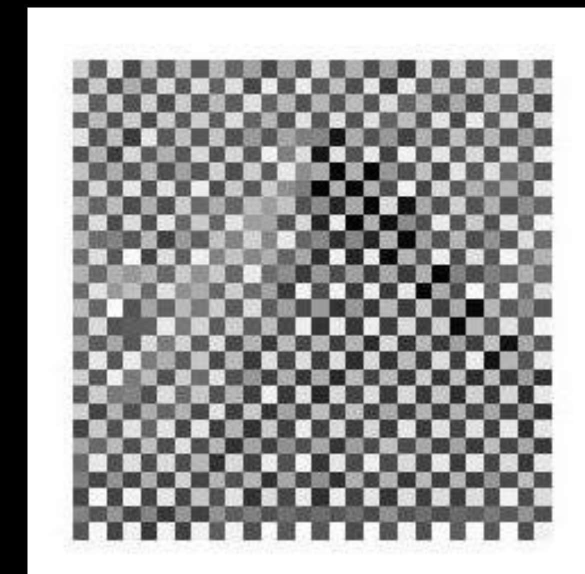
We define the distance d_{ij} between two tensors T_{ij} and $\tilde{T}_{ij}$ as the Frobenius Norm of the difference between the two tensors:

$$d_{ij} = \|T_{ij} - \tilde{T}_{ij}\|_F \tag{10}$$

$$\mathrm{dist}(T, \tilde{T}) = \sum_i \sum_j d_{ij}$$

AND TO SEARCH INDEPENDENTLY OF THICKNESS OF EDGE, WE NORMALISE THE TENSOR MATRIX.

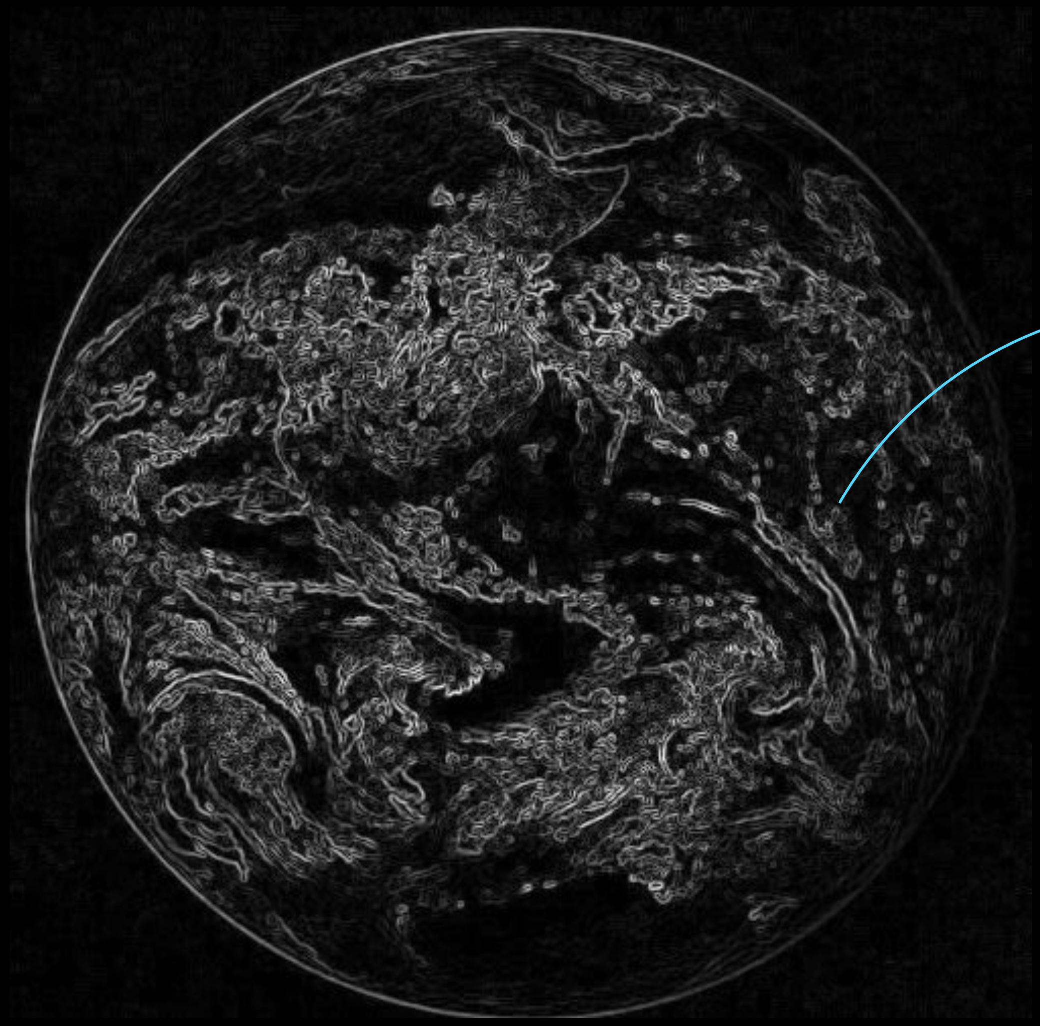
OUR APPROACH



THE ENTIRE PIPELINE (USING IMGRADIENT DIRECTLY ON RGB IMAGE) SHOWN FOR ONE OF THE IMAGES.

POSSIBLE PITFALL

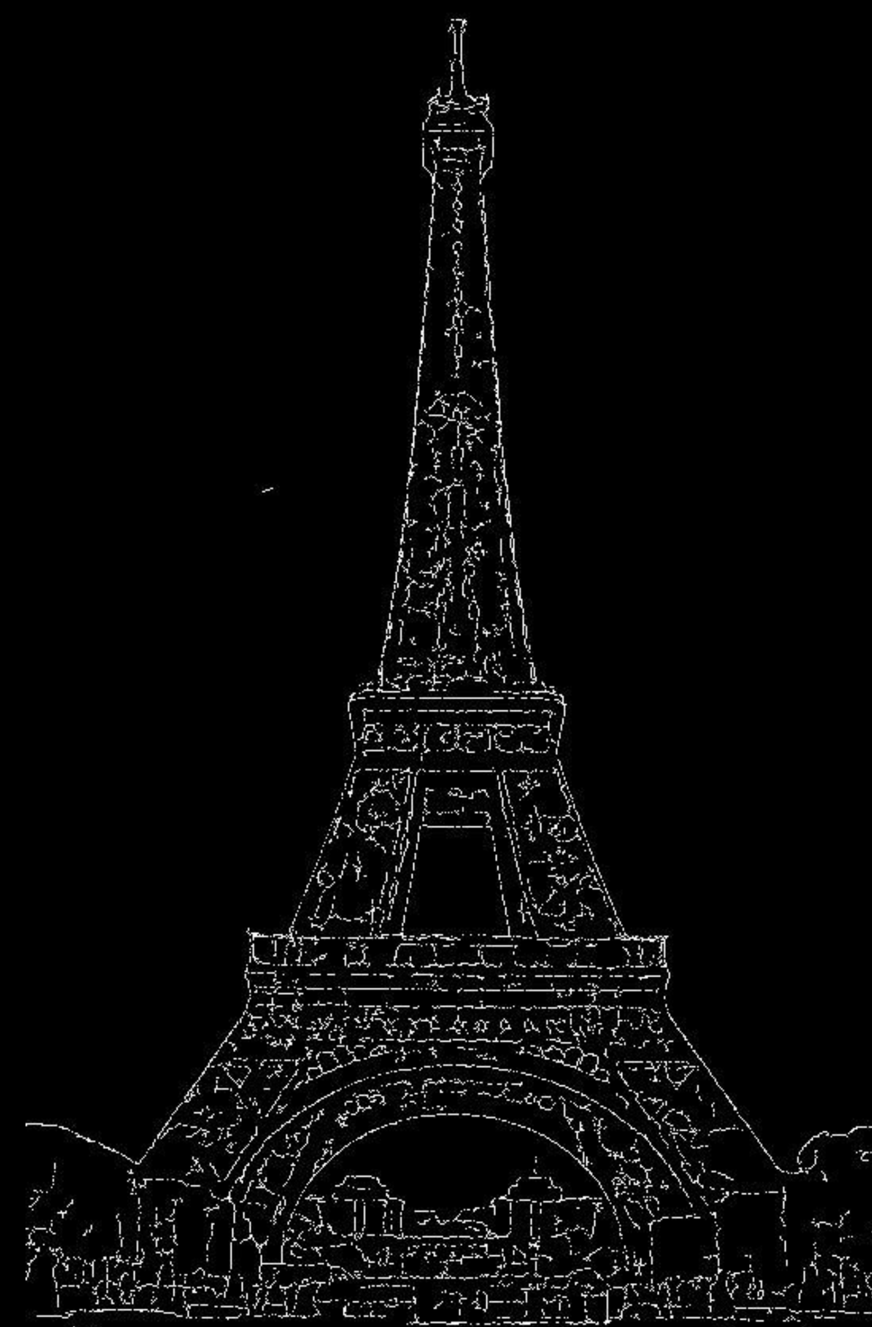
COMPLEX INTERNAL DETAILS AND BACKGROUND OBJECTS ARE PRESENT AT TIMES IN THE IMAGE GENERATED BY 'IMGRADIENT'. SINCE OUR TENSOR COMPUTATION RELIES HEAVILY ON THE PRINCIPAL EDGES IN THE IMAGES, WE WANT OUR IMAGES TO BE AS 'CLEAN' AS POSSIBLE (AND AT THE SAME TIME ALSO RETAIN THE GENERAL OUTER SHAPE OF THE OBJECT).



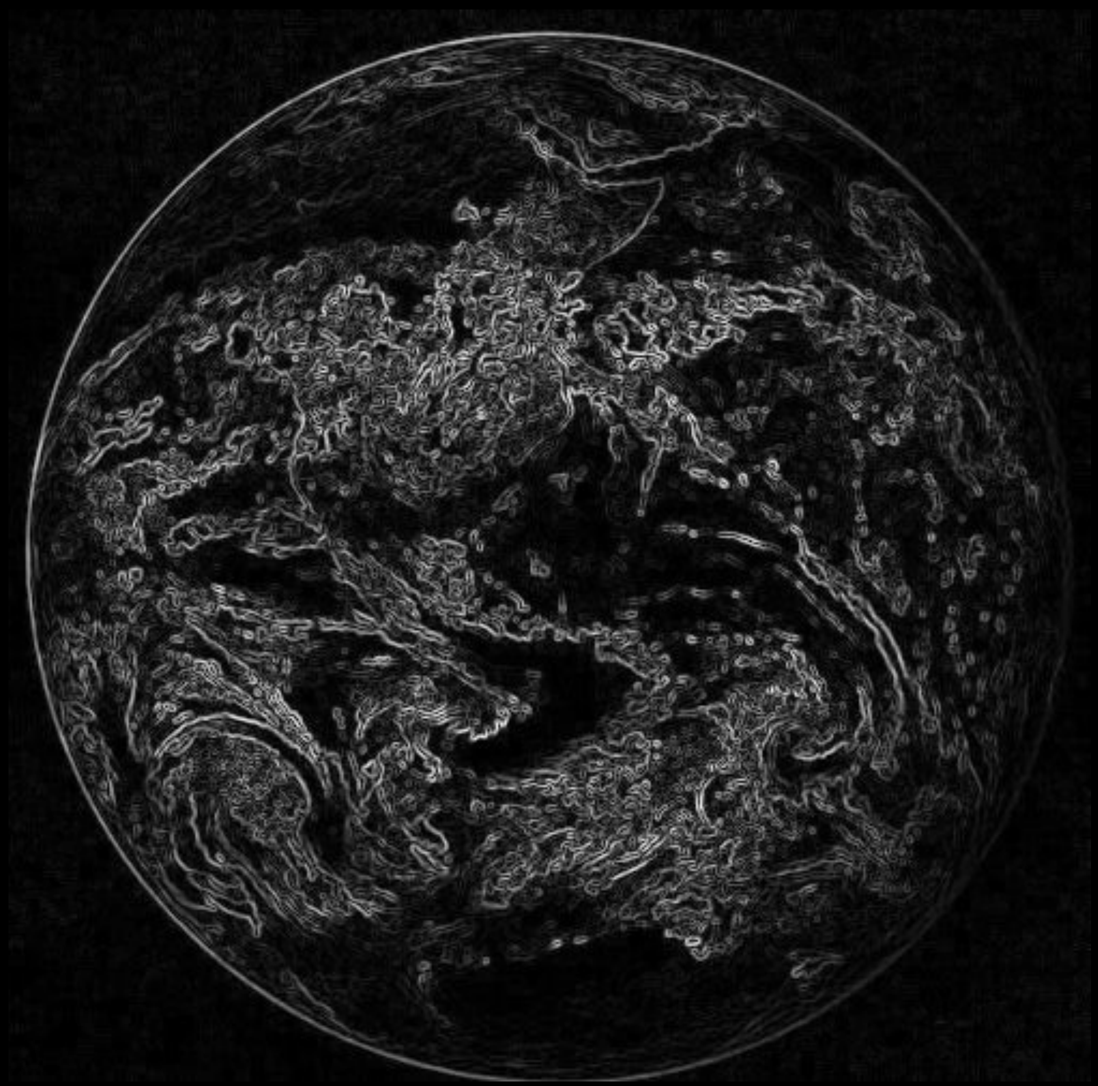
These complex details have non-zero contribution in the tensor vector and reduce accuracy

OUR SOLUTION

INSTEAD OF USING GRADIENT IMAGE OF RGB IMAGE, WE GENERATE SKETCH TOKENS FOR THE IMAGE AND COMPUTE GRADIENT OF IT. THEN WE COMPUTE TENSOR FROM THIS SKETCH-TOKEN-IMAGE.

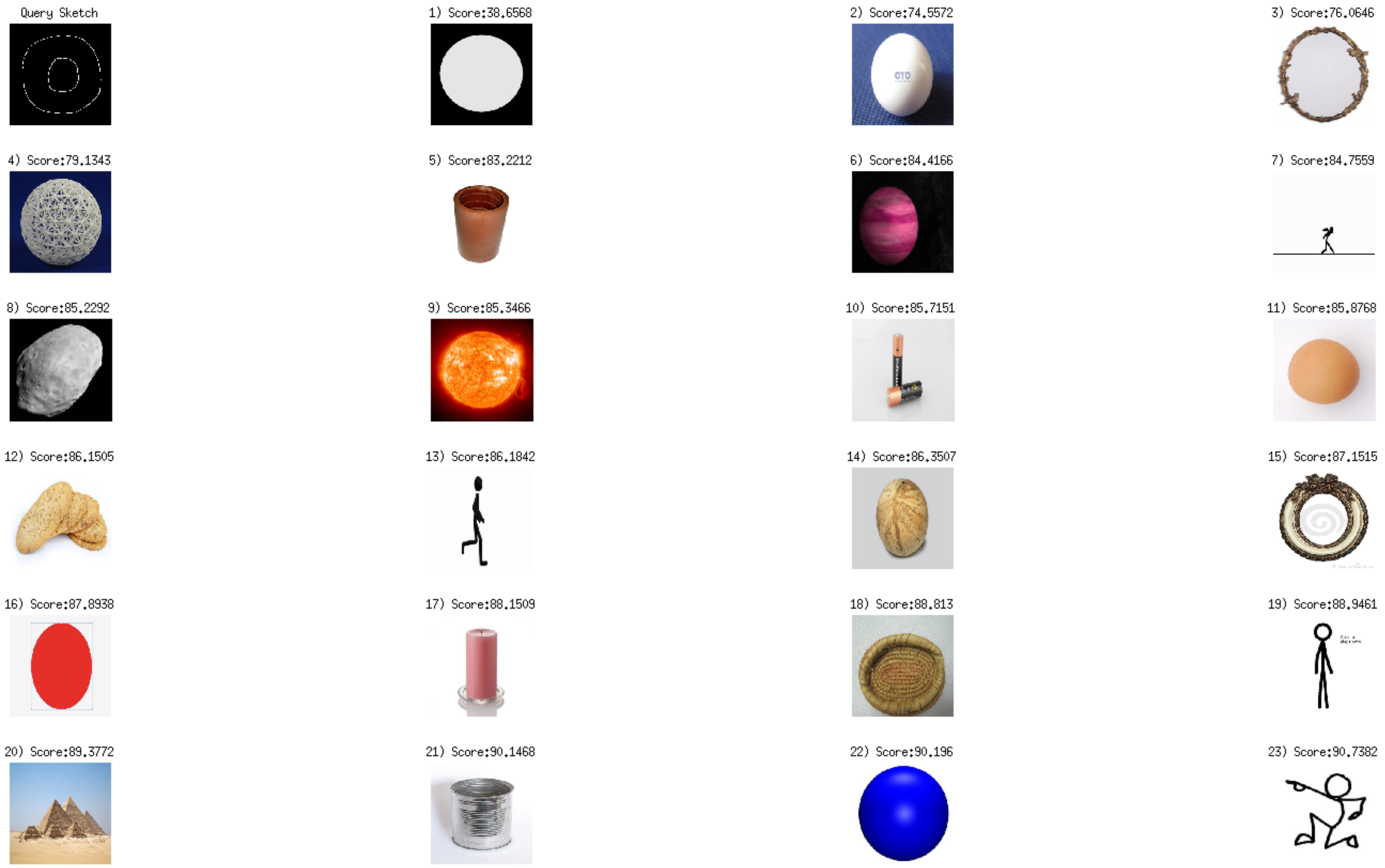


SKETCH TOKENS



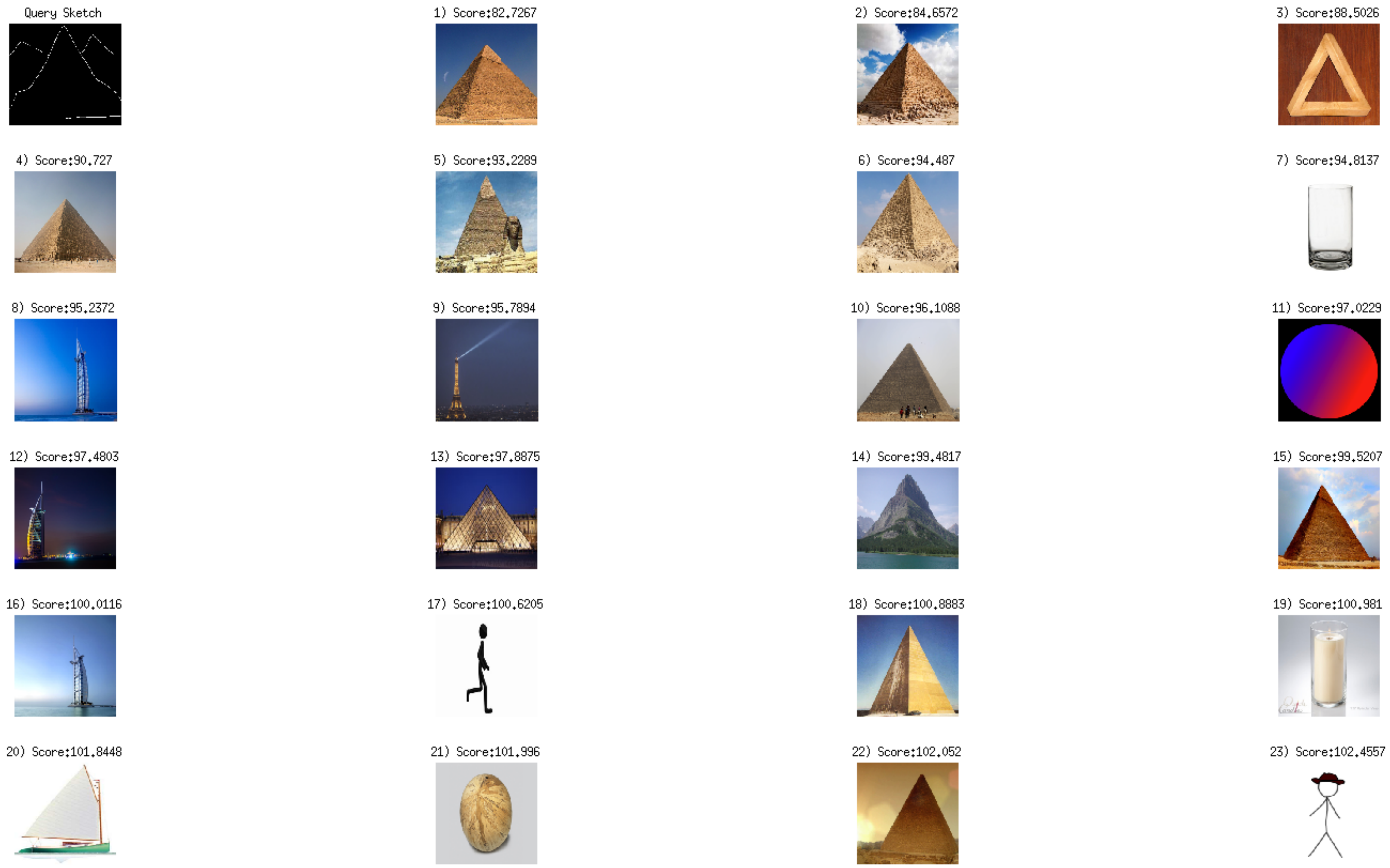
THE GRADIENT IMAGES OBTAINED FROM RGB (ON LEFT) AND SKETCH-TOKEN (ON RIGHT) IMAGES.

RESULTS



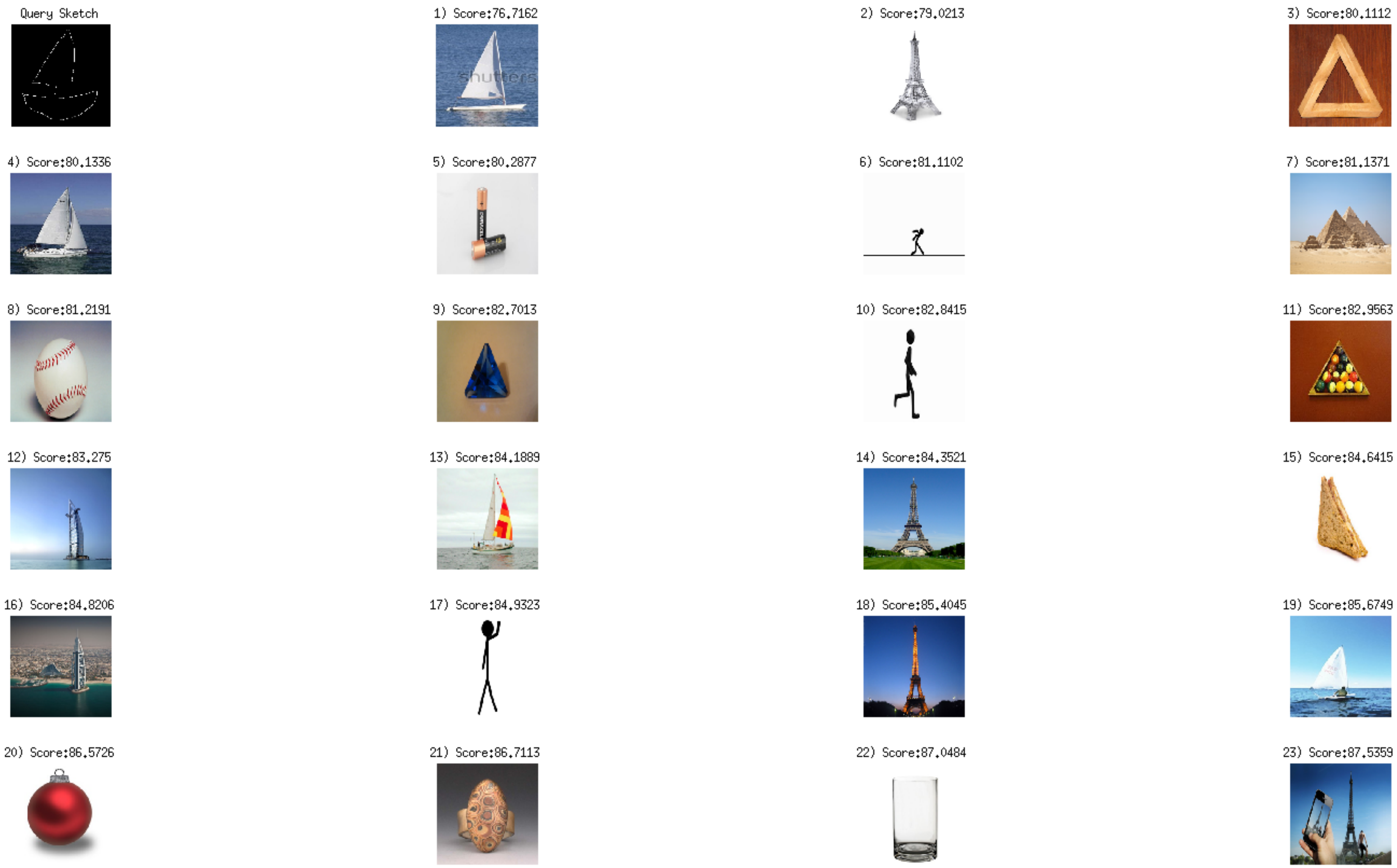
TOP LEFT IMAGE IS THE QUERY IMAGE. SEARCH RESULTS ARE RANKED BASED ON DISTANCE AND THE FIRST 23 RESULTS ARE DISPLAYED IN INCREASING ORDER OF DISTANCE (FROM THE QUERY SKETCH) IN LEFT-RIGHT AND TOP-BOTTOM FASHION.

RESULTS



TOP LEFT IMAGE IS THE QUERY IMAGE. SEARCH RESULTS ARE RANKED BASED ON DISTANCE AND THE FIRST 23 RESULTS ARE DISPLAYED IN INCREASING ORDER OF DISTANCE (FROM THE QUERY SKETCH) IN LEFT-RIGHT AND TOP-BOTTOM FASHION.

RESULTS



TOP LEFT IMAGE IS THE QUERY IMAGE. SEARCH RESULTS ARE RANKED BASED ON DISTANCE AND THE FIRST 23 RESULTS ARE DISPLAYED IN INCREASING ORDER OF DISTANCE (FROM THE QUERY SKETCH) IN LEFT-RIGHT AND TOP-BOTTOM FASHION. MANY OF THE IMAGES SEEM LIKE A MISMATCH, BUT IN REALITY, THEIR SHAPES ARE VERTICAL AND SO IS THE SHAPE OF THE SAIL AND MAST IN THE SKETCH.

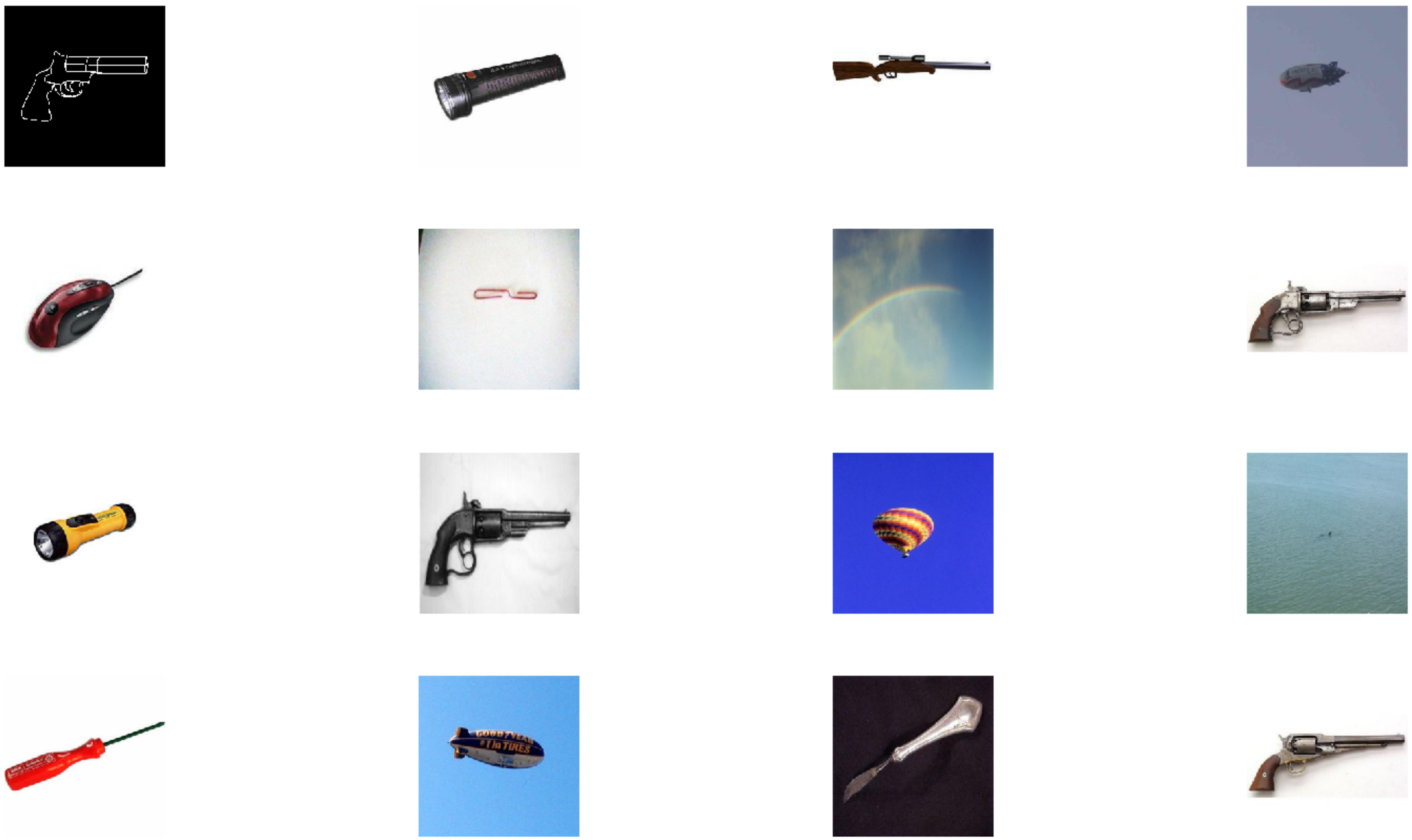
RESULTS

TOP LEFT IMAGE IS THE QUERY IMAGE. SEARCH RESULTS ARE RANKED BASED ON DISTANCE AND THE FIRST 15 RESULTS ARE DISPLAYED IN INCREASING ORDER OF DISTANCE (FROM THE QUERY SKETCH) IN LEFT-RIGHT AND TOP-BOTTOM FASHION. THESE IMAGES ARE FROM THE TU-BERLIN DATASET AND WERE USED FOR THE INITIAL DATABASE. OBSERVE THAT BOTH THE CUP-IMAGE AND ITS MIRROR IMAGE ARE RETURNED.

RESULTS

TOP LEFT IMAGE IS THE QUERY IMAGE. SEARCH RESULTS ARE RANKED BASED ON DISTANCE AND THE FIRST 15 RESULTS ARE DISPLAYED IN INCREASING ORDER OF DISTANCE (FROM THE QUERY SKETCH) IN LEFT-RIGHT AND TOP-BOTTOM FASHION. THESE IMAGES ARE FROM THE TU-BERLIN DATASET AND WERE USED FOR THE INITIAL DATABASE.

RESULTS

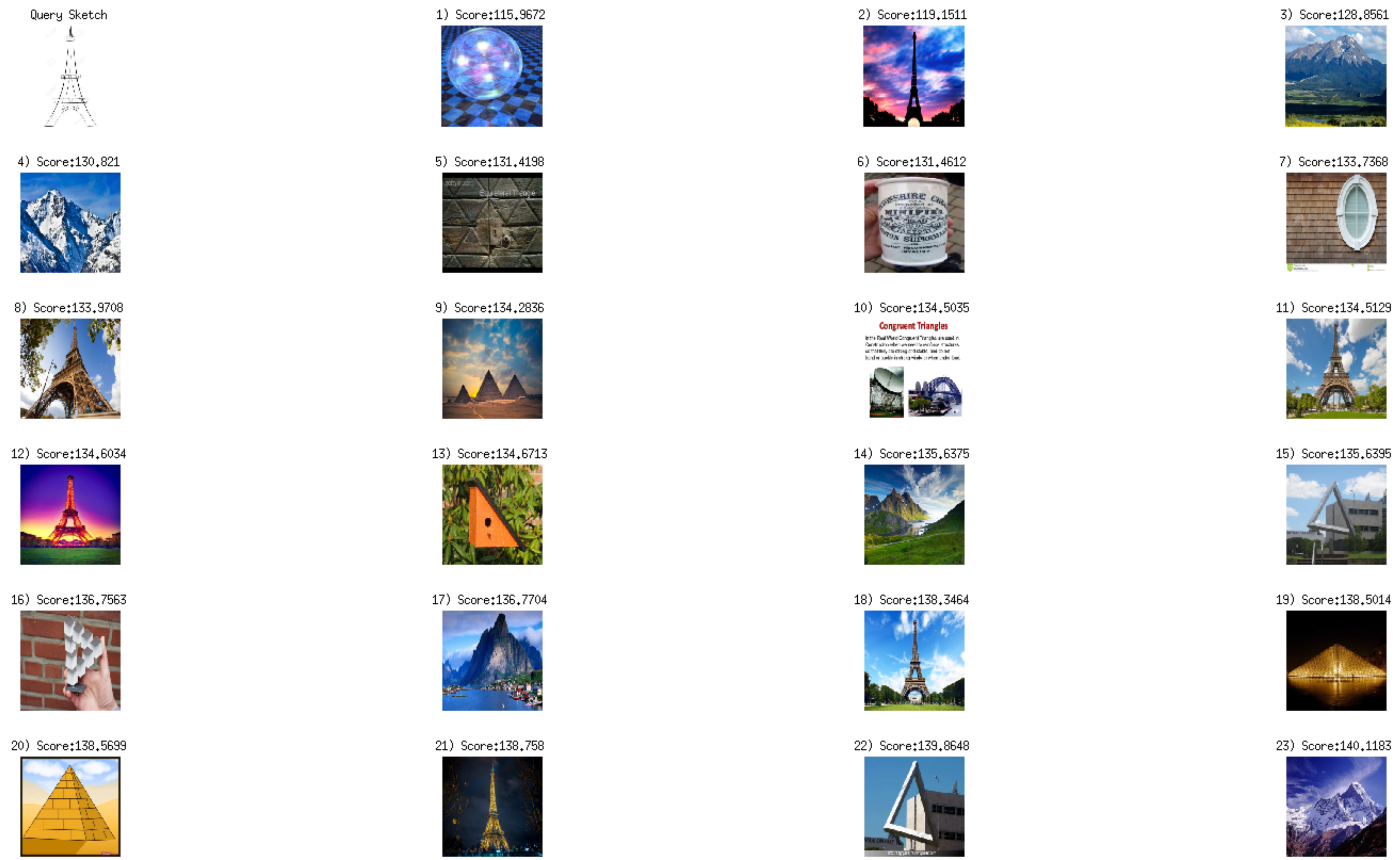


TOP LEFT IMAGE IS THE QUERY IMAGE. SEARCH RESULTS ARE RANKED BASED ON DISTANCE AND THE FIRST 15 RESULTS ARE DISPLAYED IN INCREASING ORDER OF DISTANCE (FROM THE QUERY SKETCH) IN LEFT-RIGHT AND TOP-BOTTOM FASHION.

RESULTS (BEFORE SKETCH-TOKEN)

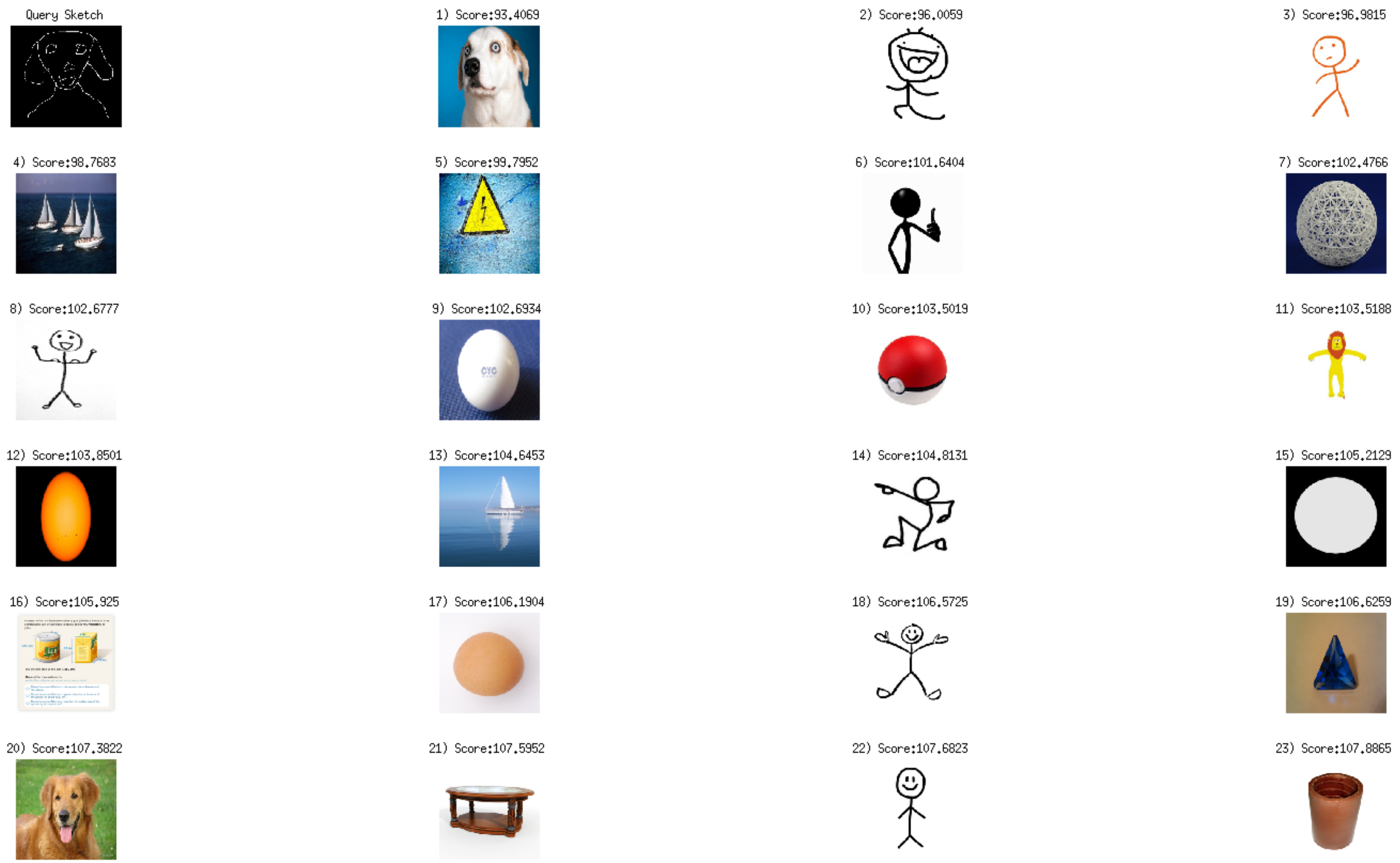
TOP LEFT IMAGE IS THE QUERY IMAGE. SEARCH RESULTS ARE RANKED BASED ON DISTANCE AND THE FIRST 23 RESULTS ARE DISPLAYED IN INCREASING ORDER OF DISTANCE (FROM THE QUERY SKETCH) IN LEFT-RIGHT AND TOP-BOTTOM FASHION.

RESULTS (AFTER USING SKETCH-TOKENS)



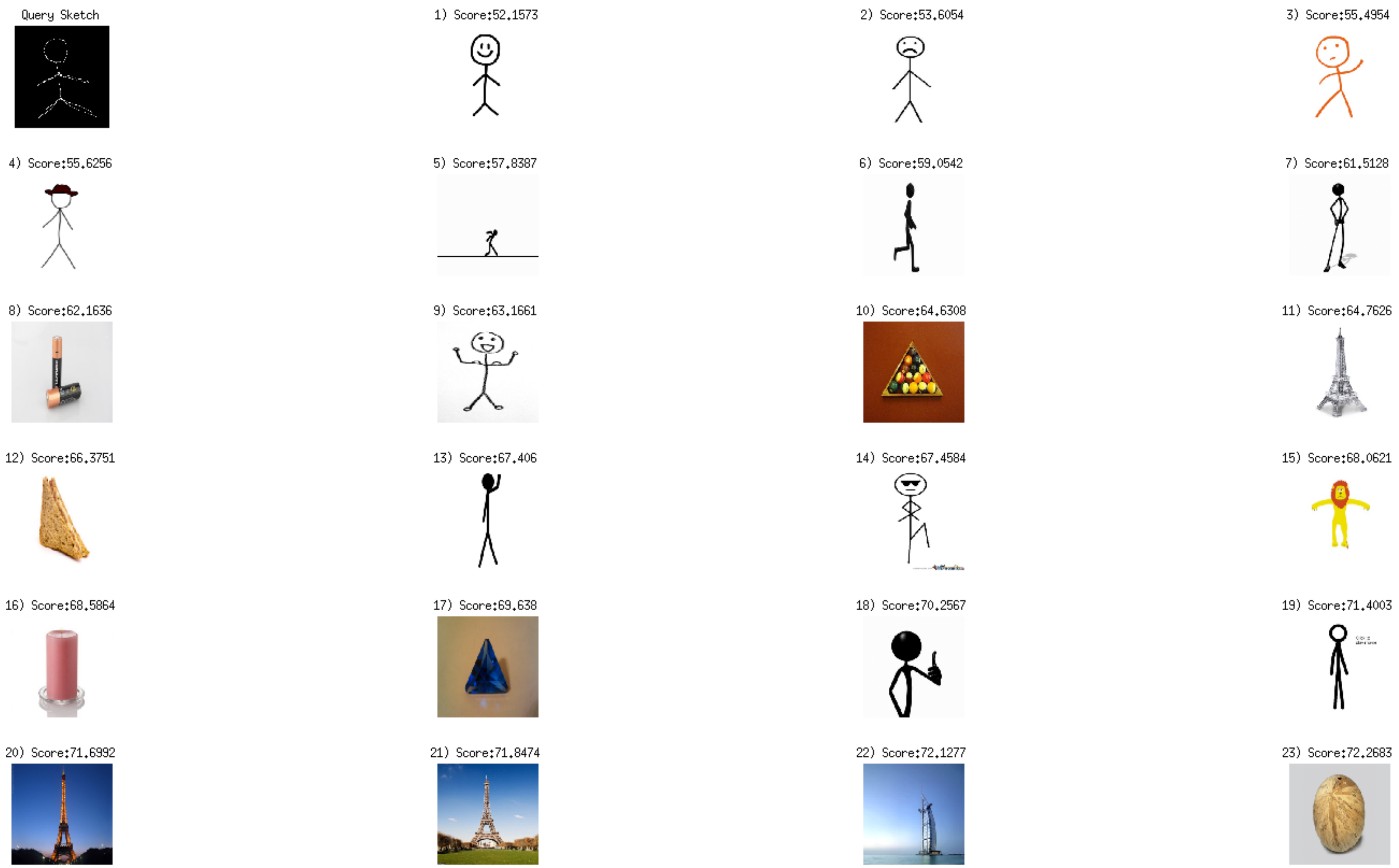
TOP LEFT IMAGE IS THE QUERY IMAGE. SEARCH RESULTS ARE RANKED BASED ON DISTANCE AND THE FIRST 23 RESULTS ARE DISPLAYED IN INCREASING ORDER OF DISTANCE (FROM THE QUERY SKETCH) IN LEFT-RIGHT AND TOP-BOTTOM FASHION.

RESULTS



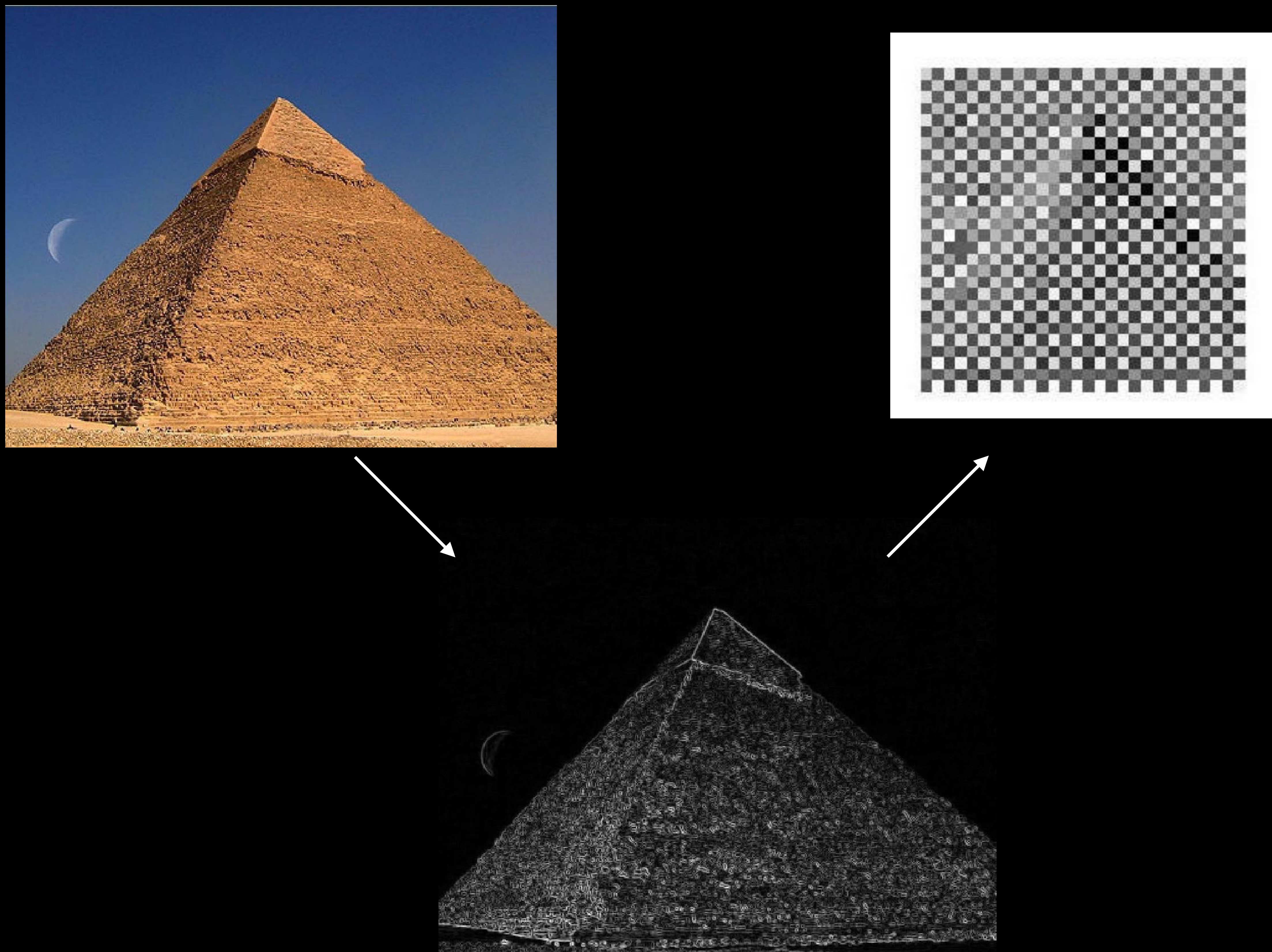
TOP LEFT IMAGE IS THE QUERY IMAGE. SEARCH RESULTS ARE RANKED BASED ON DISTANCE AND THE FIRST 23 RESULTS ARE DISPLAYED IN INCREASING ORDER OF DISTANCE (FROM THE QUERY SKETCH) IN LEFT-RIGHT AND TOP-BOTTOM FASHION.

RESULTS

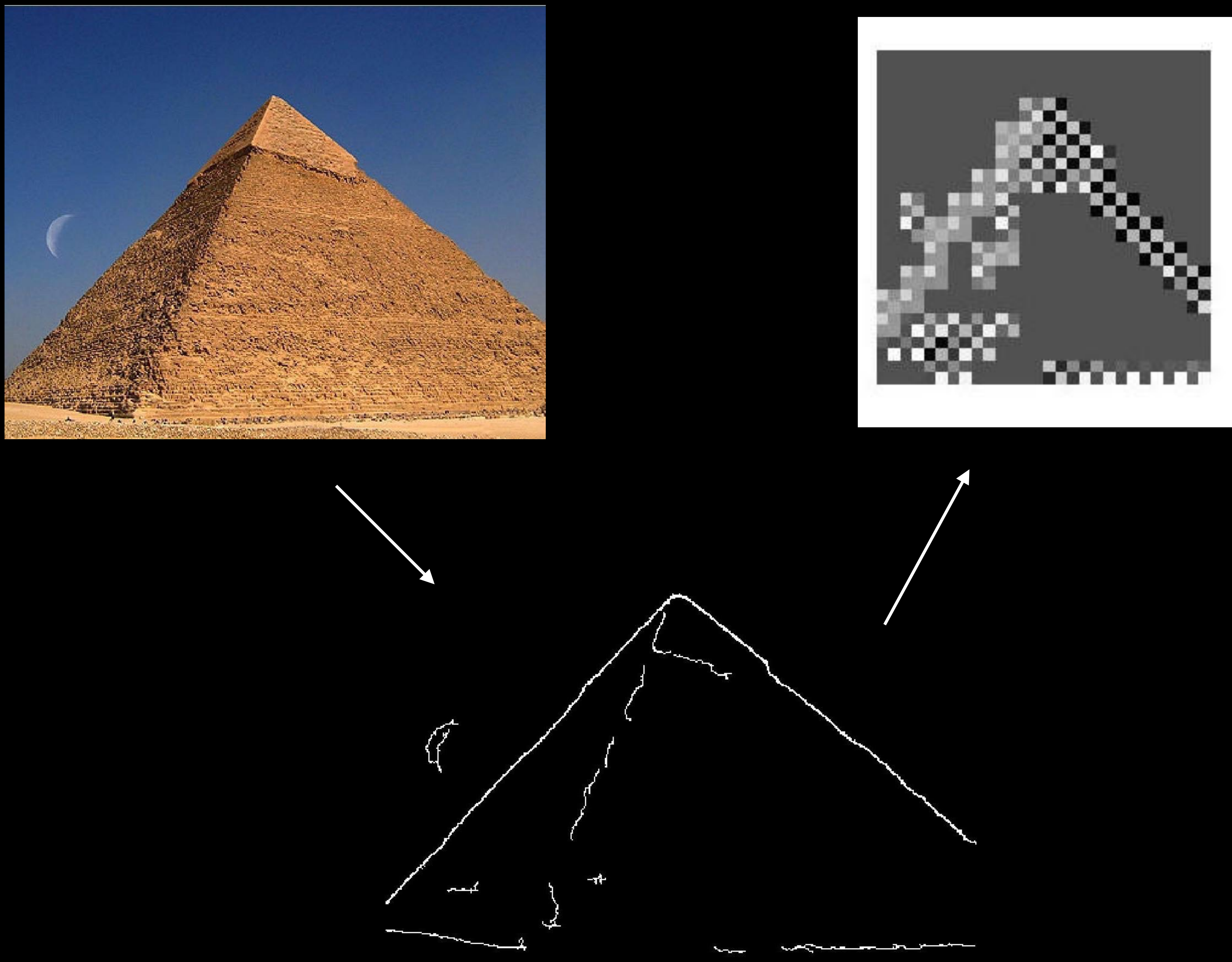


TOP LEFT IMAGE IS THE QUERY IMAGE. SEARCH RESULTS ARE RANKED BASED ON DISTANCE AND THE FIRST 23 RESULTS ARE DISPLAYED IN INCREASING ORDER OF DISTANCE (FROM THE QUERY SKETCH) IN LEFT-RIGHT AND TOP-BOTTOM FASHION.

USING RGB IMAGE DIRECTLY



USING SKETCH TOKENS IMAGE



DRAWBACKS/SCOPE FOR IMPROVEMENT

OUR APPROACH DOES NOT TAKE SEMANTIC INFO INTO CONSIDERATION. IT BASICALLY DOES SHAPE MATCHING. FOR EXAMPLE, A BMW CAR AND A TRUCK HAVING SAME ORIENTATION WILL BE TREATED SIMILAR TO EACH OTHER WHILE AN IMAGE OF THE SAME BMW BUT IN A DIFFERENT ORIENTATION MIGHT NOT BE RETURNED.

IT DEPENDS ON RELATIVE POSITION IN FINAL IMAGE. BUT THIS MIGHT ACTUALLY BE CONSIDERED AS AN ADVANTAGE BECAUSE WE RETURN IMAGES WHICH WILL LOOK VERY CLOSE TO SKETCH AND THIS MIGHT BE WHAT IS DESIRED BY THE USER.

REFERENCES

'A DESCRIPTOR FOR LARGE SCALE IMAGE RETRIEVAL BASED ON SKETCHED FEATURE LINES' BY MATHIAS EITZ, ET AL (EUROGRAPHICS 2009).

'SKETCH TOKENS: A LEARNED MID-LEVEL REPRESENTATION FOR CONTOUR AND OBJECT DETECTION' BY JOSEPH J. LIM, ET AL (CVPR 2013).

'HOW DO HUMANS SKETCH' BY MATHIAS EITZ, ET AL (SIGGRAPH 2012).

Thank You!