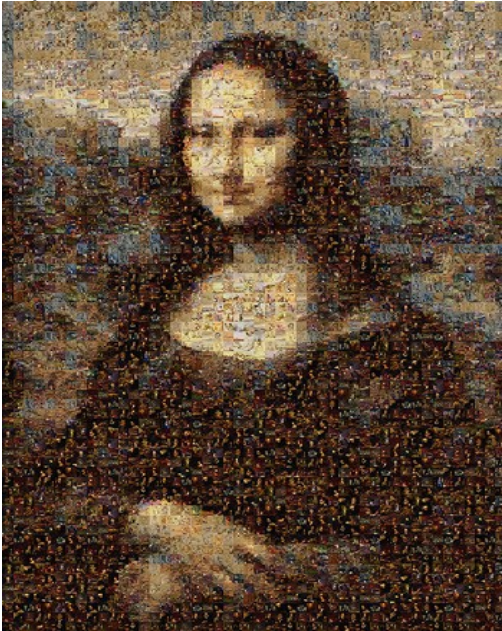



Title	PhotoMosaic
<p>Description</p>	<p>It was once a famous advertisement for Coca-cola. It was a picture of the Coca-cola logo. When zooming out, you see the Coca-cola logo. But if you zoom in, you find that this logo is actually composed of thousands of face photos.</p> <p>Wikipedia has the following description for a PhotoMosaic: “In the field of photographic imaging, a photographic mosaic, also known under the term Photomosaic, a portmanteau of photo and mosaic, is a picture (usually a photograph) that has been divided into (usually equal sized) rectangular sections, each of which is replaced with another photograph that matches the target photo.^[1] When viewed at low magnifications, the individual pixels appear as the primary image, while close examination reveals that the image is in fact made up of many hundreds or thousands of smaller images.”</p> <p>And here are two examples (the smaller images might not be clear in this small printout, but you can find many such examples on the Internet):</p> <div style="display: flex; justify-content: space-around; align-items: center;">   </div> <p>This project idea is about implementing a tool that gets an image, and generates a PhotoMosaic for it. Mainly this tool should maintain a large collection of images (several tens of thousands), and use them in creating the mosaics. Given an input image, the tool should draw an imaginary grid over it, compute some color feature for every grid cell (I suggest to compute the <i>dominant color</i> using a <i>color coherence vector</i>), find from the collection an image having a similar color feature, and place it in this grid cell in the output mosaic. Doing this, one expects to get such a nicely looking</p>



	PhotoMosaic. Be careful, the devil is in the details! The idea is simple, but there is a lot of fine tuning and performance issues there.
Group size	3 – 5 members.
Duration	6 weeks.
Deliverables	1- A program that gets an image, and outputs its PhotoMosaic. The image collection used for creating the mosaics should be controllable by the user. That is, the user should be able to build his own collection and/or edit the existing collection. 2- A short (2 pages) user manual + a technical document describing the used techniques.
Bonus extensions	- Maintain the image collection using a DBMS, and use some <i>open database connectivity</i> APIs to connect the program to the database.
Mentor	Dr. Mahmoud Attia Sakr.
Notes	<p>In order to implement this Plugin you'll need to use some image processing library (e.g., <i>cimg.sourceforge.net</i>), read about color feature extraction (e.g., [1]), and probably learn how to connect to a database from your code. C++ should be fine for implementing this program, but you are free to use whatever language you like.</p> <p>This project requires good programming and research skills. Unless you are an A+ team, it might be risky to apply for this idea.</p> <p>[1]Greg Pass, Ramin Zabih, and Justin Miller. 1997. Comparing images using color coherence vectors. In <i>Proceedings of the fourth ACM international conference on Multimedia</i> (MULTIMEDIA '96).</p>