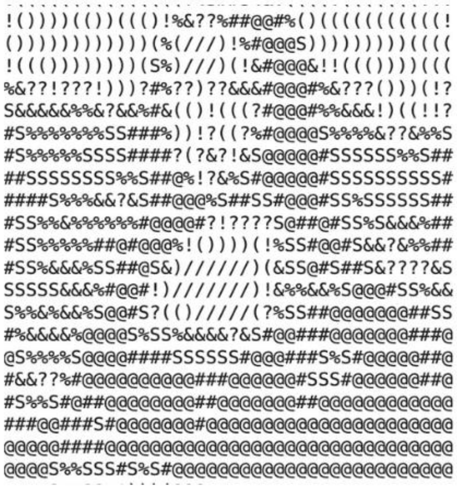
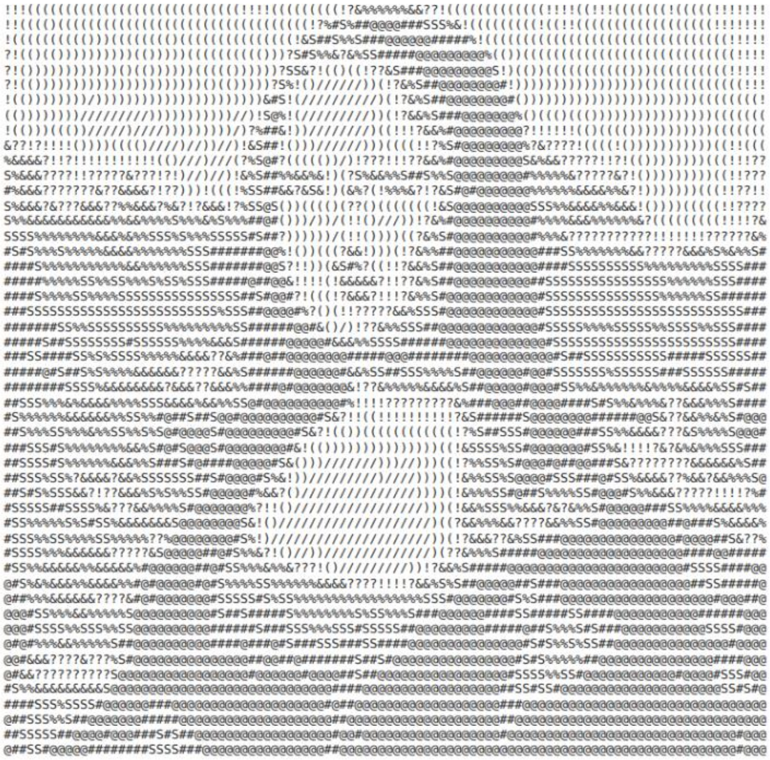
Manvi Rankawat **2024**

ASCII ART

Project showcasing ASCII art, beginning with the Mona Lisa in 40 and 100-character widths. The comparison highlights how pixel size affects detail, with the narrower version capturing basic elements and the wider version revealing more intricacies. Additional images demonstrate ASCII art's versatility in reinterpreting classic masterpieces.



```
!()))(())((()!%&??%##@@#%()(((((((((((!
())))))))))))))(%(///)!%#@@@S)))))))))((((
!((()))))))))(S%)///)(!&#@@@&!!((()))))(((
%&??!???!))))?#%??)??&&&#@@@#%&???()))(!?
S&&&&&%%&?&&%#&(()!(((?#@@@#%%&&&!)((!!?
#S%%%%%%%SS###%))!?((?%#@@@@S%%%%&??&%%S
#S%%%%%SSSS####?(?&?!&S@@@@@#SSSSSS%%S##
##SSSSSSSSS%%S##@%!?&%S#@@@@@#SSSSSSSSSS#
####S%%%&&?&S##@@@%S##SS#@@@#SS%SSSSSS##
#SS%%&%%%%%%#@@@@#?!????S@##@#SS%S&&&%##
#SS%%%%%##@#@@@%!()))))(!%SS#@@#S&&?&%%##
#SS%&&&%SS##@S&)///////)(&SS@#S##S&????&S
SSSSS&&&%#@@#!)/////////)!&%%&&%S@@@#SS%&&
S%%&%&&%S@@#S?(()//////(?%SS##@@@@@@@##SS
#%&&&&%@@@@S%SS%&&&&?&S#@@###@@@@@@@###@
@S%%%%S@@@@####SSSSSS#@@@###S%S#@@@@@##@
#&&??%#@@@@@@@@@@###@@@@@@#SSS#@@@@@@##@
#S%%S#@##@@@@@@@@@##@@@@@@@@##@@@@@@@@@@@@
###@@###S#@@@@@@@@#@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@####@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@S%%SSS#S%S#@@@@@@@@@@@@@@@@@@@@@@@@@@@
```



PROBLEM DOMAIN & PROJECT DESCRIPTION

The goal of Project2 of Computer Vision_CSCI_6527_10 is to develop an ASCII Art Generator that converts digital images into text-based visual representations.

ASCII art replaces pixel intensity with characters that have varying densities, where darker areas are mapped to denser characters (such as @ or #) and lighter areas to less dense ones (like , or .).

This transformation allows for a creative, low-resolution rendering of images using only textual characters.

The primary inputs of the system are image files (in standard formats which in our case is .jpg), and the output is a text file containing the ASCII representation of the image.

For example, an input could be a photograph of the Mona Lisa, and the output would be a text file with ASCII characters arranged to resemble the painting.

This tool is valuable to artists who enjoy creating retro or minimalist artwork, and for developers seeking lightweight visual representations in environments with text-only display constraints, such as command-line interfaces.

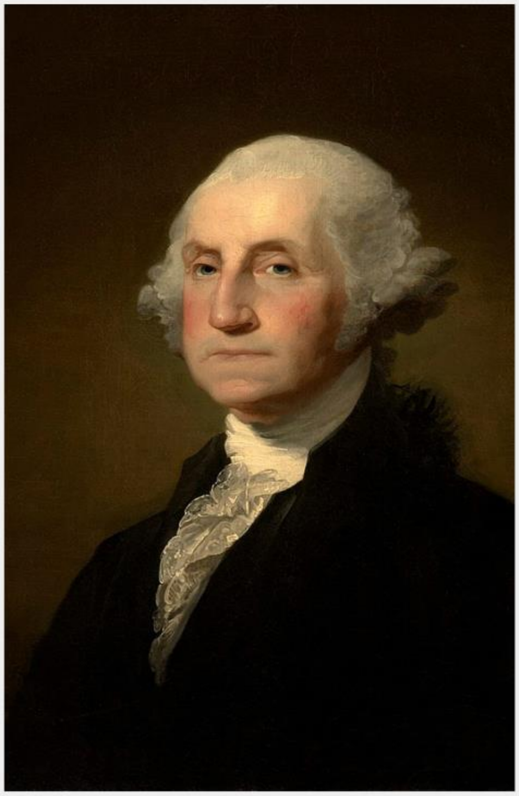
ASCII_Art.ipynb

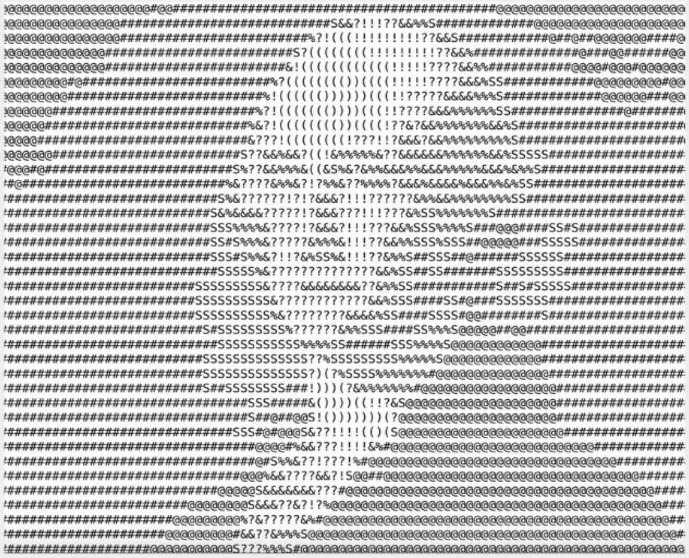
Approach:

The high-level approach is as follows:

Input Image Processing: The input image is first resized to reduce its resolution while maintaining the original aspect ratio. This ensures that the output ASCII art maintains the visual proportions of the original image.
Grayscale Conversion: The image is converted into a grayscale format to simplify the mapping process, as each pixel is now represented by a brightness level between 0 (black) and 255 (white).

Mapping to ASCII Characters: Each pixel's brightness is mapped to a corresponding ASCII character from a predefined set based on the intensity. Darker pixels correspond to characters like @ or #, while lighter pixels are mapped to characters like . or ,
Output Formatting: The ASCII characters are arranged in lines corresponding to the original image's width to ensure the text forms a coherent representation of the image.





```
@@@@@@@@@@@@@@@@@#@@#########################################@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@###########################S&&?!!!??&&%%S###########@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@########################%?!(((!!!!!!!!!??&&S############@##@##@@@@@@@####@
@@@@@@@@@@@@@########################S?((((((((!!!!!!!!!??&&%################@###@@######@@
@@@@@@@@@@@@##########################&!((((((((((((!!!!!????&&%%###########@@@@#@@@#@@@@@@
@@@@@@@@#@#########################%?((((((())((((!!!!!????&&&%SS############@@@@@@@@#@@
@@@@@@@@##########################%!((((((()))))))(((!!?????&&&&%%%S#############@@@@@@###@@
@@@@@@##########################%?!((((((()))))(((!!????&&&%%%%%%SS################@######
@@@@@#########################%&?!(((((((())((((!??&?&&%%%%%%%&&%S######################
@@@@##########################&???!(((((((((!???!!?&&&?&&%%%%%%%%%S######################
@@@@@@######################S??&&%&&?((!&%%%%%&??&&&&&&%%%%%%&&%SSSSS#################
@@@#@######################S%??&&%%%&((&S%&?&%%&&&%%&&&%%%%%&&&%&%%S#################
#@########################%&????&%%&?!?%%&??%%%%?&&&%&&&&%&&&%%&%SS##################
##########################S%&??????!?!?&&&?!!!??????&%%&&%%%%%%%%%SS####################
#########################S&%&&&&?????!?&&&???!!!???&%SS%%%%%%%S######################
#########################SSS%%%%&????!?&&&?!!!???&&%SSS%%%%S###@@@####SS#S###############
#########################SS#S%%%&?????&%%%&!!!??&&%%SSS%SSS##@@@@@###SSSSS###############
#########################SSS#S%%&?!!?&%SS%&!!!??&%%S##SSS##@######SSSSSS################
##########################SSSSS%&?????????????&&%SS##SS#######SSSSSSSSS################
########################SSSSSSSSS&????&&&&&&&&??&%%SS###########S##S#SSSSS###############
########################SSSSSSSSSS&??????????&&%SSS####SS#@###SSSSSSS################
########################SSSSSSSSSS%&????????&&&&%SS####SSSS#@@########S##################
#########################S#SSSSSSSSS%??????&%%SSS####SS%%%S@@@@@##@@####################
##########################SSSSSSSSSSS%%%%SS######SSS%%%%S@@@@@@@@@@@@###################
#########################SSSSSSSSSSSSSS??%SSSSSSSSS%%%%%S@@@@@@@@@@@@@@###################
#########################SSSSSSSSSSSSSS?)(?%SSSS%%%%%%%#@@@@@@@@@@@@@@@@##################
#########################S##SSSSSSSS###!))))(?&%%%%%%%#@@@@@@@@@@@@@@@@@@################
###############################SSS#####&()))))((!!?&S@@@@@@@@@@@@@@@@@@@@################
###############################S##@##@@S!())))))))(?@@@@@@@@@@@@@@@@@@@@@################
#############################SSS#@#@@@S&??!!!!(()(S@@@@@@@@@@@@@@@@@@@@@@################
################################@@@@#%&&???!!!!&%#@@@@@@@@@@@@@@@@@@@@@@@@@@@###########
################################@#S%%&??!???!%#@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@########
##############################@@@%&&????&&?!S@@##@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@######
#########################@@@@@S&&&&&&&???#@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@####
#######################@@@@@@@@S&&&??&?!?%@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@###
#####################@@@@@@@@@%?&?????&%#@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@##
####################@@@@@@@@@@#&&??&%%%S@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@#
###################@@@@@@@@@@@@S???%%%S#@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
```

Datasets and Challenges:

Took Painting references as I feel this could be useful for beginners in art-world to refers to ASCII characters and consider painting based on ASCII characters as in which part could be dark or light on facial features.
Dataset of Painting of Mona Lisa, George Washington, Girl with a Pearl Earing and Indian Prime Minister Mr. Narendra Modi is pulled.

Images with high contrast tend to work better, as the ASCII characters can more easily represent these differences.
Apart from which I experimented the ASCII image with varying height and width of 40, 60 & 100 and as a result observed that the Image with high size-range has more clarity which is the same case as more the pixel better is the clarity.

Maintaining the correct aspect ratio is important so that the ASCII art doesn't become distorted. This is handled by adjusting the height-to-width ratio of the resized image.

```
@@@@@@@@@@@@@@@@@@@###!)%##?)(!?&&&&?????!?&%SS###########################@&(%&###&&###@@@@@@@@@@
@@@@@@@@@@@@@##@@@#@S!(&S%?)(?&&&&?!((%SSS%&&%SSSS#############@#########@@@&(&?###%&S####@@@@@@@@@
@@@@@@@@@@@@@@@@@@##!(?!!!()(&&&&?!)/)S@###SSSSSSSS#########################@@%(&?###%%%#####@@@@@@@@
@@@@@@@@@@@@@@@#@#@S))(((())?&&&?!!!!!&%%SS%%%%%%SSS###################@#@@@S(&?&##%%%S####@@@@@@@@
@@@@@@@@@@@@@@@@@#@S)(()()(?&&&?!!!!!?&&%&????&&SSSS###################@@@@@#(??!S#S&SS####@@@@@@@@
@@@@@@@@@@@@@@@@@@@%((((((?&%%%&!!!!!!!!!!!!???&%SS#SSS##################@@@##!!&(S##&S##S##@@@@@@@@
@@@@@@@@@@@@@@@@@@@S(((((?&%%%%&!!(!!!!!!!!!??&%%S###SS##############@@##@@###?!&(%##&%##S#@@@@@@@@@
@@@@@@@@@@@@@@@@@@@#!((((?&%%%SS&!!((!!!!!!!??&&%S##SSSSS#############@@@@@####?!&(%##&?###S#@@@@@@@@
@@@@@@@@@@@@@@@@@@#@&)((?%%%S%%SS?!!!!!!!!???&&%SS#SSSSSSSS##########@@@@@@###@&!&(&##%!S##S##@@@@@@@
@@@@@@@@@@@@@@@@@@@@S())(?S#SSSSS%!!!!!!!?&&&%SSS#SSSSSSSS###SS###@@@@@@@@@@@@%!&!&S#%)?###%#@@@@@@@
@@@@@@@@@@@@@@@@@@@@#!())(&%%%&&?!!!!????&&%%SSSSSSSSSSS#SSSS###@@@@@@@@@@@@@%(?(?S#&)(S##S%@@@@@@@
@@@@@@@@@@@@@@@@@@@@@%((!((?&&&????!!????&&%%%SSSSSSSSSS#######@@@@@@@@@@@@@@@@&)()?S#%))&###%S@@@@@@
@@@@@@@@@@@@@@@@@@#@##?(&%%%%%%&&&&????&&&%%%SSSSSSSSSSS#######@@@@@@@@@@@@@#@&)))?SSS))(###%%#@@@@@
@@@@@@@@@@@@@@@@@@@@@@@#!!%%%%&%SSS%&???&&%%%%SS%%%SSSSSS#########@@@@@@@@@@@@@@@&(((!%SS())%##S%S#@@@@
@@@@@@@@@@@@@@@@@@@@@@##%!?!!?%%%S%&????&%%%%%S%%%%%%%SSS#&!S###@@@@@@@@@@@@#@%))((%%S!()?##S%S#@@@@
@@@@@@@@@@@@@@@@@@@@@@#@?(!?&%%%%%&&??&&%%%%%%%%%%%%%%%SS#SS?(%####@@@@@@@@##@##S)(!(&&S?()(###S&#@@@@
@@@@@@@@@@@@@@@@@@@@@@@#@S))((!&&%%&&&&&%%%%%%%%%%%%%%SSSSSSSS%SS#@@##############!!?!??S%())S##S&%@@@@
@@@@@@@@@@@@@@@@@@@@@@@@%((((!?&&%%&%%%%%%%%%%%%%%SSSSSSSSS#SSSSS%%&&&&&&&%SSSS#?!?!&(&S!))%##SS&S@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@S&?!!?&&%%%%%%%%%%%%%SSSSSSSSSSS%&?!!!(((!????&&%SSS#&!?!&(!S&))?##SS%%@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@##S%SSSS####%&%%%%%%%S%%&??(())////))(!?&%SSSS######?(?!&!(S%()(###SS%#@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@#?!??&&&!(/////////)(!?%S###############&!!!!&!&S!))S##SS%%@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@#&!!())//////)!?&%S##################@S?!?(??&S&()&##SS%&#@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@%()/)////)(?%%SS###################@#?!?(!&?S%!(!###%S&S@
```

RESULTS

(ASCII_ARTGENERATOR.IPYNB FILE)

The results are surprisingly effective for simple, high-contrast images like portraits, logos, or famous artworks (e.g., Mona Lisa or Girl with a Pearl Earring). When compared to alternative approaches (e.g., using more complex graphical algorithms for ASCII art), this method achieves a balance between simplicity and visual quality.

It is effective for text-based environments but could struggle with more complex, color-rich images. This approach solves the problem of rendering simple images as ASCII art for casual or artistic use.

However, for real-world use cases such as image compression or more detailed visual representations, the ASCII art generator would need improvements, such as using a larger set of characters or adding color support.

TOOLS AND RESOURCES

This project was implemented using Python, with the Pillow library (pip install pillow) used to handle image processing.

Pillow is an essential tool for this type of project as it provides easy functions for resizing, converting images to grayscale, and accessing pixel data.

The logic for converting pixel brightness to ASCII characters was inspired by classic ASCII art techniques used in early computer systems, where graphics capabilities were limited.

The core idea behind the project is simple but effective: by using different characters to represent different brightness levels, you can create a rough but recognizable visual representation of an image using just text.

Overall, this project shows how simple tools like Python and ASCII characters can be combined to produce interesting and creative results, offering a fun way to transform images into something new.