GURU NANAK COLLEGE BUDHLADA

DEPARTMENT: COMPUTER

NAME OF PROJECT: image To ASCII

Submitted to:

HOD

Dr. Rekha Kalra

Submitted by:

Yaman (321825)

Yash(321824)

Chushamdeep Singh(321850)

Table of Contents

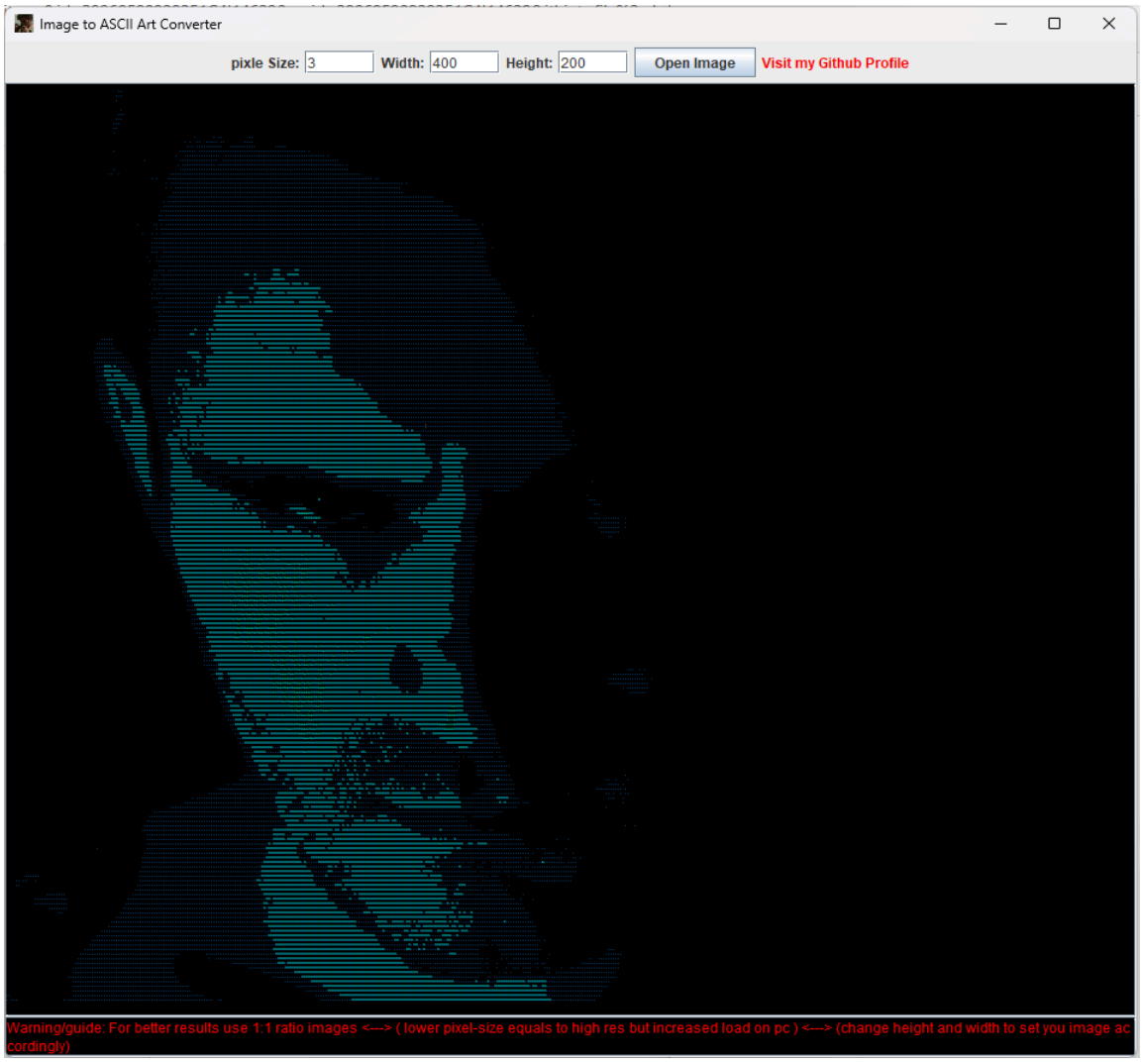
1. Introduction

Image to ASCII Art Converter converts images into ASCII art. It has a graphical user interface (GUI).

- open an image file
- adjust settings
    - pixel size
    - Width-Height
- And view the resulting ASCII representation of the image.

Project repository (https://github.com/dullat)

Preview: 1

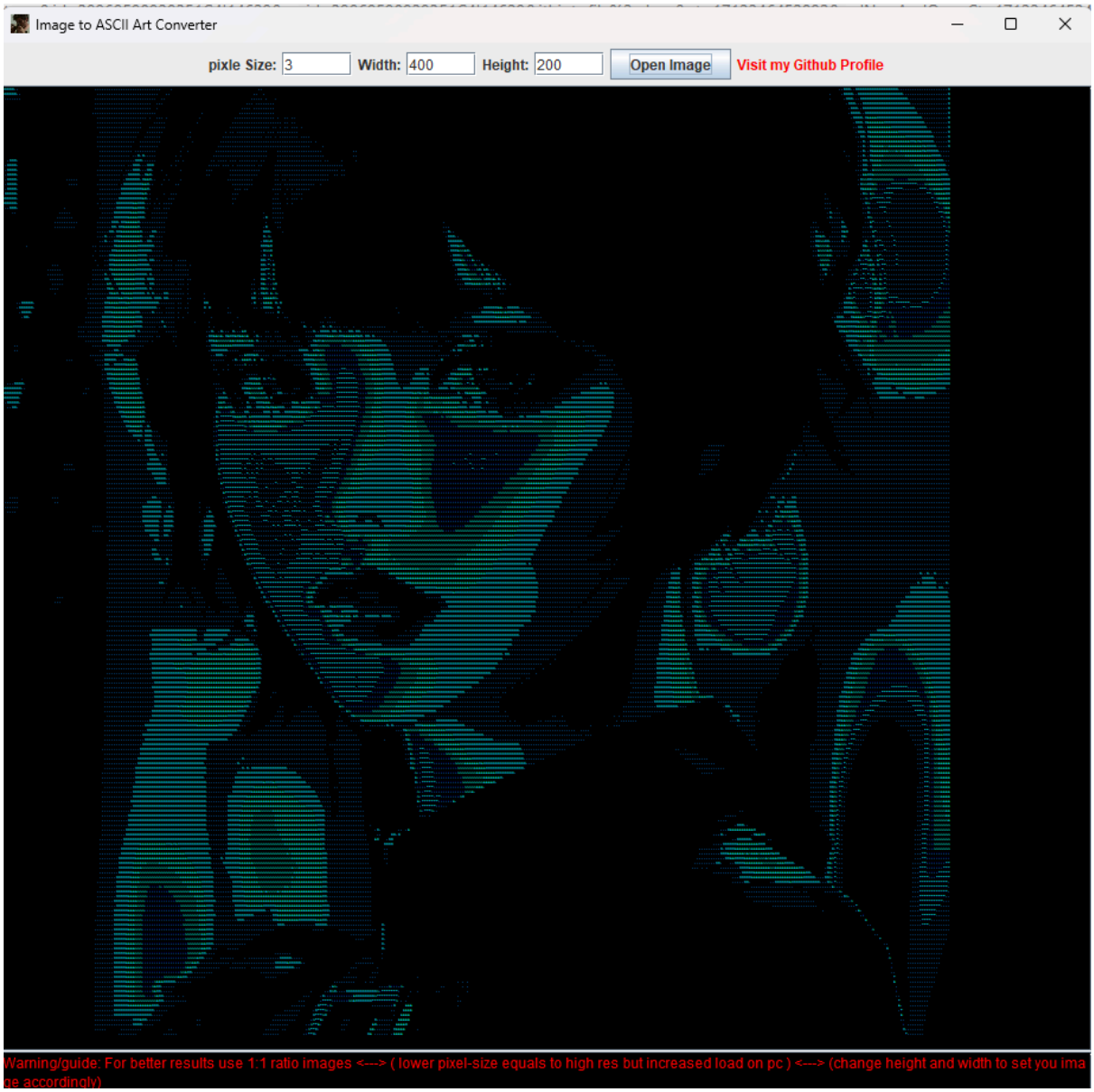


2. Functionality

The main functionality of the application includes:

- Opening an image file using a file chooser dialog.
- Adjusting pixel size, width, and height of the output ASCII art.
- Displaying the ASCII representation of the image in a text area.
- Providing warnings and guidance for optimal results.

Preview 2:



3. Code Explanation

Class Structure

- **ImageToAsciiGUI**: The main class representing the GUI application. It extends **JFrame**.

Methods

- **openImage()**: Opens a dialog to select an image file, reads the selected image and generates ASCII art.
- **updateFontSize()**: Updates the font size of the ASCII art.
- **generateAsciiArt**: Generates ASCII art from a image.
- **resize**: Resizes image to a specified width and height.
- **mapToAscii(int gray)**: Maps a grayscale value to an ASCII character.
- **displayMessage(String message)**: Displays a message in the message area of the GUI.
- **clearMessage()**: Clears the message area.
- **main(String[] args)**: Entry point . invoke GUI

CODE

```
import javax.imageio.ImageIO;

import javax.swing.*;


import java.awt.*;

import java.awt.image.BufferedImage;

import java.io.File;

import java.io.IOException;

import java.net.URI;
```

```
import java.awt.event.*;



public class ImageToAsciiGUI extends JFrame {
    private JTextArea asciiTextArea;
    private JTextField fontSizeTextField;
    private JTextField widthTextField;
    private JTextField heightTextField;
    private JTextArea messageArea;

    public ImageToAsciiGUI() {
        setTitle("Image to ASCII Art Converter");
        setSize(800, 600);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JPanel contentPane = new JPanel();
        contentPane.setLayout(new BorderLayout());
        contentPane.setBackground(Color.BLACK);

        asciiTextArea = new JTextArea();
        asciiTextArea.setEditable(false);
        asciiTextArea.setFont(new Font("Monospaced", Font.PLAIN, 3));// default pixl size to 3
        asciiTextArea.setForeground(Color.CYAN); // setting color to aqua
        asciiTextArea.setBackground(Color.BLACK);// setting bg
        JScrollPane scrollPane = new JScrollPane(asciiTextArea);

        // pixelsize label and text field
```

```
JLabel fontSizeLabel = new JLabel("pixle Size:");
fontSizeTextField = new JTextField("3", 5); // Default font size
fontSizeTextField.addActionListener(e -> updateFontSize());



JLabel widthLabel = new JLabel("Width:");
widthTextField = new JTextField("400", 5); // Default width
JLabel heightLabel = new JLabel("Height:");
heightTextField = new JTextField("200", 5); // Default height

//image button
JButton openImageButton = new JButton("Open Image");
openImageButton.addActionListener(e -> openImage());

// control panel
JPanel controlPanel = new JPanel();
controlPanel.add(fontSizeLabel);
controlPanel.add(fontSizeTextField);
controlPanel.add(widthLabel);
controlPanel.add(widthTextField);
controlPanel.add(heightLabel);
controlPanel.add(heightTextField);
controlPanel.add(openImageButton);

//warnings
messageArea = new JTextArea("Warning/guide: For better results use 1:1 ratio images <---> ( lower pixel-size equals to high res but increased load on pc ) <---> (change height and width to set you image accordingly)");
messageArea.setEditable(false);
```

```
        messageArea.setForeground(Color.RED);
        messageArea.setBackground(Color.BLACK);
        messageArea.setLineWrap(true);

        JScrollPane messageScrollPane = new JScrollPane(messageArea);



        contentPane.add(scrollPane, BorderLayout.CENTER);
        contentPane.add(controlPanel, BorderLayout.NORTH);
        contentPane.add(messageScrollPane, BorderLayout.SOUTH);

        setContentPane(contentPane);
        setVisible(true);

        // setting logo
        try {
            BufferedImage iconImage = ImageIO.read(getClass().getResourceAsStream("profile.png"));

//            BufferedImage iconImage = ImageIO.read(new File("profile.png"));
            setIconImage(iconImage);
        } catch (IOException ex) {
            System.out.println("cant load image: " + ex.getMessage());
        }

        //github link
        JLabel githubLink = new JLabel("Visit my Github Profile");
        githubLink.setCursor(new Cursor(Cursor.HAND_CURSOR));
        githubLink.setForeground(Color.RED);
```

```
    githubLink.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent e) {
            try {
                Desktop.getDesktop().browse(new URI("https://github.com/dullat"));
            } catch (Exception ex) {
                ex.printStackTrace();
            }
        }
    });
    controlPanel.add(githubLink);

}

private void openImage() {
    JFileChooser fileChooser = new JFileChooser();
    int returnValue = fileChooser.showOpenDialog(this);

    if (returnValue == JFileChooser.APPROVE_OPTION) {
        File selectedFile = fileChooser.getSelectedFile();
        try {
            BufferedImage image = ImageIO.read(selectedFile);
            String asciiArt = generateAsciiArt(image);
            asciiTextArea.setText(asciiArt);
            //clearMessage();// clear warning message. disabled
        } catch (IOException ex) {
            ex.printStackTrace();
            displayMessage("Error: selectedfile is not valid.");
```

```
        }
    }
}

private void updateFontSize() {
    try {
        int fontSize = Integer.parseInt(fontSizeTextField.getText());
        asciiTextArea.setFont(new Font("Monospaced", Font.PLAIN, fontSize));
        //clearMessage(); // Clear any previous messages
    } catch (NumberFormatException ex) {
        displayMessage("enter a valid font size.");
    }
}

private String generateAsciiArt(BufferedImage image) {
    StringBuilder asciiArt = new StringBuilder();



    int asciiWidth = Integer.parseInt(widthTextField.getText());
    int asciiHeight = Integer.parseInt(heightTextField.getText());

    BufferedImage resizedImage = resize(image, asciiWidth, asciiHeight);

    for (int y = 0; y < asciiHeight; y++) {
        StringBuilder asciiRow = new StringBuilder();
        for (int x = 0; x < asciiWidth; x++) {
            int pixel = resizedImage.getRGB(x, y);
            int gray = (getRed(pixel) + getGreen(pixel) + getBlue(pixel)) / 4;
```

```
            char asciiChar = mapToAscii(gray);
            asciiRow.append(asciiChar);
        }
        asciiArt.append(asciiRow).append("\n");
    }
    return asciiArt.toString();
}

private BufferedImage resize(BufferedImage image, int width, int height) {
    BufferedImage resizedImage = new BufferedImage(width, height,
BufferedImage.TYPE_INT_RGB);
    Graphics2D g = resizedImage.createGraphics();
    g.drawImage(image, 0, 0, width, height, null);
    g.dispose();
    return resizedImage;
}

private int getRed(int rgb) {
    return (rgb >> 16) & 0xFF;
}

private int getGreen(int rgb) {
    return (rgb >> 8) & 0xFF;
}

private int getBlue(int rgb) {
    return rgb & 0xFF;
}
```

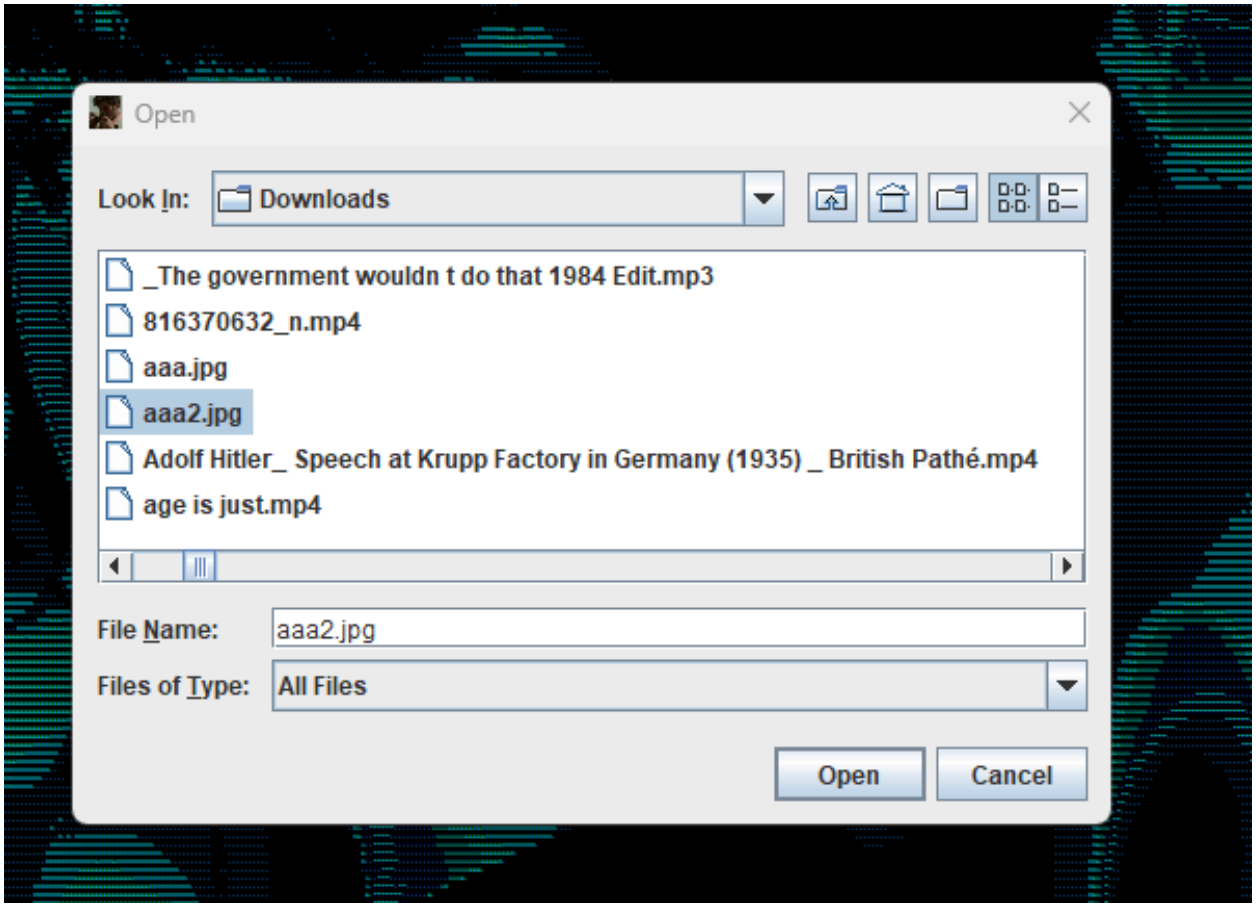
```
    private char mapToAscii(int gray) {
        char[] asciiChars = {' ', '.', '8', '&', 'o', ':', '*', '.', ' '}; // ASCII characters working as Pixles
        int index = (int) (gray * ((asciiChars.length - 1) / (255.0 * 0.7)));
        return asciiChars[index];
    }



    private void displayMessage(String message) {
        messageArea.setText(message);
    }

    private void clearMessage() {
        messageArea.setText("");
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(ImageToAsciiGUI::new);
    }
}
```

4. Usage Guide

To use the Image to ASCII Art Converter:

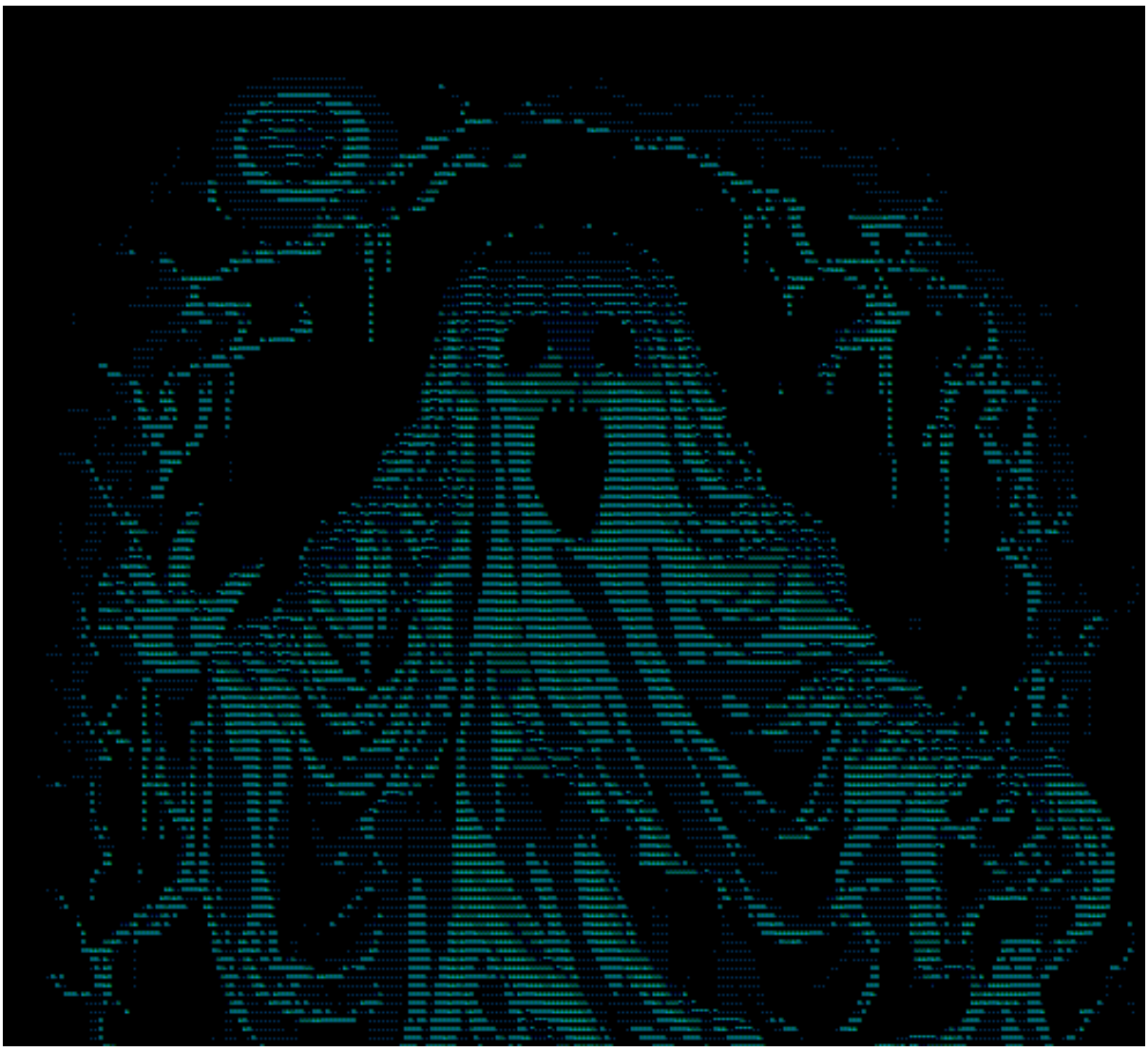
- Launch the application by running the main method.
- Click on the "Open Image" button to select an image file.
- Adjust the pixel size, width, and height settings as desired.
- View the resulting ASCII art in the text area.
- For optimal results, follow the provided warnings and guidance.

Choose an image

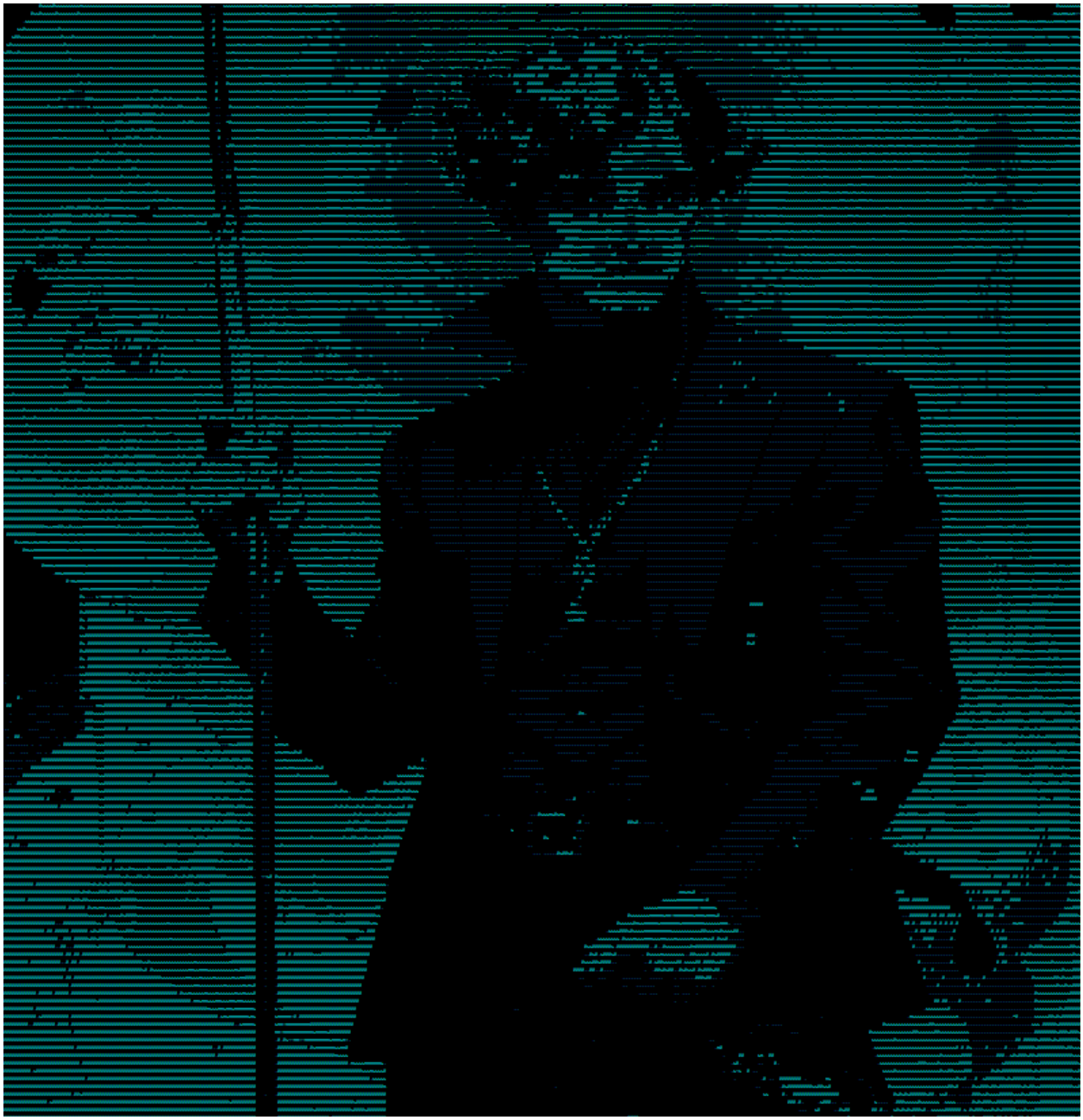


5. OUTPUT

5.1 OUPTUT 1



5.2 OUTPUT 2



6 . Conclusion

Image to ASCII Art Converter is a simple, creative and powerful tool to convert images into ASCII art, it is a creative way to visualize digital images. This app has very simple but user-friendly interface, allowing users to convert their images into ASCII art. App provide different attributes to manipulate art generation. .exe and project is on my Github.